Context-Aware Surveillance Video Summarization

Shu Zhang, Yingying Zhu, and Amit K. Roy-Chowdhury

***Abstract*— We present a method that is able to find the most informative video portions, leading to a summarization of video sequences. In contrast to the existing works, our method is able to capture the important video portions through information about individual local motion regions, as well as the interactions between these motion regions. In particular, our proposed context-aware video summarization (CAVS) framework adopts the methodology of sparse coding with generalized sparse group lasso to learn a dictionary of video features and a dictionary of spatiotemporal feature correlation graphs. Sparsity ensures that the most informative features and relationships are retained. The feature correlations, represented by a dictionary of graphs, indicate how motion regions correlate with each other globally. When a new video segment is processed by CAVS, both dictionaries are updated in an online fashion. In particular, CAVS scans through every video segment to determine if the new features along with the feature correlations can be sparsely represented by the learned dictionaries. If not, the dictionaries are updated, and the corresponding video segments are incorporated into the summarized video. The results on four public data sets, mostly composed of surveillance videos and a small amount of other online videos, show the effectiveness of our proposed method.**

***Index Terms*— Video summarization, context, sparse coding.**

I. Introduction

THE huge growth in video data calls for an urgent need to develop tools to summarize events occurring in these videos. Large parts of most videos are often redundant or not informative. Manually watching hours of videos to figure out the informative events is very time consuming. Furthermore, it is difficult for people to focus on watching videos for hours and not miss important events in the video. So, it is very important to develop tools that allow analysts to automatically select the most informative parts of a video sequence. The problem of finding such informative video portions is usually considered as the problem of video summarization.

Although the video summarization problem has been extensively studied, many previous methods worked on structured videos [1], e.g. sports videos and movies. These videos have well-organized structures which can be exploited in the summarization process, but may not be applicable in other natural videos. In recent studies, the video summarization problem has been often defined as the problem of feature reconstruction [2], [3]. This is essentially to determine if the features in the test dataset can be reconstructed by those in the summarized data. However, they have not considered the fact that objects and events are often inter-related to each other, which can be very efficiently exploited in the summarization process. Such relationships can often be observed in videos, especially surveillance videos. This inter-relationship, often termed as *context* information, has been very effective for many object and activity recognition problems [4], [5]. This paper explores this aspect from the perspective of the surveillance video summarization problem.

Many videos consist of complex events that have strong correlations between each other. For instance, Fig. 1 shows three scenarios from surveillance videos and user-generated videos. Some informative events are highlighted, which are expected to be summarized. The first scenario from a surveillance video shows that a person in white gets out of the car, closes the door and leaves the car. In the second scenario, the person gets out of the car, walks to the back of the car and opens the trunk. In the third scenario from a user generated video, three kids collect leaves, stand up and throw leaves to others. According to the problem formulation in existing works like [2], both the first two scenarios have the event of getting out of the car and such an event may not be shown in all the summarized videos. However, analysts may want to watch summarized videos as stories or series of informative events. Similarly, in the third scenario, analysts do not just want to focus on a single activity such as collecting the leaves. Instead, they would like to watch the whole series of activities. Rather than watching a short yet non-informative event such as entering a car, video watchers can be more interested in watching a slightly longer but informative summarized video sequence , e.g., what a person does after getting out of the car. The importance of the correlations between different events is obviously of significance and should be considered in determing the summary.

In this work we consider the spatio-temporal correlations between events to be as important as the events themselves. Thus, we propose a surveillance video summarization framework that is able to find new events as well as different event correlations. When we select an informative video portion as part of the summarization results, the new events, along with the spatio-temporal correlation between them, are learned. This makes our proposed method significantly different from previous related works [2], [6], [7]. We term this as Context-Aware Video Summarization (CAVS), a framework

Manuscript received May 7, 2015; revised November 20, 2015, June 16, 2016, and August 10, 2016; accepted August 10, 2016. Date of publication August 18, 2016; date of current version September 28, 2016. This work was supported by NSF under Grant IIS 1316934. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Philippe Thiran.

S. Zhang and Y. Zhu were with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: shu.zhang@am.sony.com; yyzhu@google.com).

A. K. Roy-Chowdhury is with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: amitrc@ece.ucr.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes a summarized video along with the original video sequence. The total size of the video is 4.75 MB. Contact szhan011@ucr.edu for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2601493

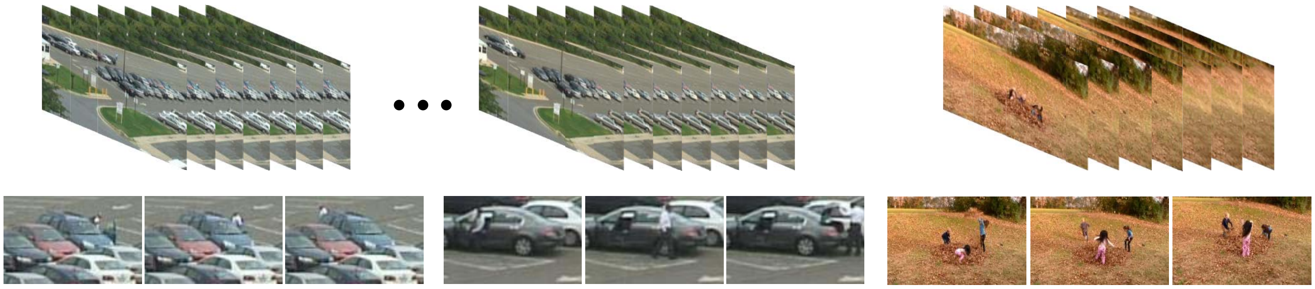


Fig. 1. Examples of video segments deemed as important by our summarization framework. From left to right: (1) from getting out of a vehicle to leaving a car, (2) from getting out of a vehicle to opening a trunk, and (3) from collecting leaves to throwing leaves to others. In the first two scenarios, although the same event, getting out of car happens in both cases, the other events that happen around it may determine that it is important enough to be summarized in both cases. In the third scenario, although each single activity appears for multiple times, the series of activities are of interest to users. Also, by summarizing the entire segments, rather than individual events, CAVS produces a more meaningful output.

that incorporates the event correlations to generate a short video summarizing the most informative parts of a long video sequence. The sparse representation, a method to represent high-dimensional samples using less training data, is adopted in CAVS to guarantee that the size of the summarized video is as small as possible.

During the training phase, the video features that describe events, e.g. spatio-temporal motion features, are first extracted. CAVS learns a dictionary of these features, summarizing the main contents of the training videos. Besides, the spatio-temporal correlations between features are also learned, represented by a dictionary of feature correlation graphs. The learned representative training features are used to sparsely reconstruct the features in the testing data using the generalized sparse group lasso [8], [9]. Specifically, a video sequence in the testing dataset is divided into segments, each of which may contain multiple events or motions. CAVS scans through every video segment along time. The new features in a new video segment are compared with the known ones in each detected region, as well as the inter-relationships between them. If the features as well as their correlations in a new video segment can be sparsely represented by the learned features, this video segment is assumed to be non-informative. Otherwise, the new video segment indicates that some important unseen information occurs in this video segment and should be absorbed into the summarized video. The corresponding features are added into the learned dictionary, and the new feature correlations are also updated in the dictionary of correlation graphs. This process is performed online until every video segment is scanned by the algorithm. We demonstrate the effectiveness of our algorithm on two state-of-the-art surveillance video datasets [10], [11] and two user-generated online videos [12], [13]. Each video in these datasets contain multiple events that interact with each other in space and time. It is demonstrated in Sec. IV that our proposed method outperforms three state-of-the-art video summarization approaches [2], [7], [14].

A. Contributions

We summarize our main contributions as follows.

- We propose a novel framework to find the most informative parts of a video sequence. Our proposed model preserves the correlations between the motion regions and therefore is able to preserve the global motion information. The spatio-temporal correlations between different events are represented by a dictionary of spatio-temporal feature correlation graphs.
- The video summarization problem is formulated as the problem of sparse feature reconstruction. This is achieved through the generalized sparse group lasso, and ensures that only the most informative portions of the video are selected.
- We propose a method for online updating of the feature dictionary and the dictionary of feature correlation graphs.
- We demonstrate the effectiveness of our algorithm on two public surveillance datasets and two user generated datasets that contain multiple spatio-temporal events.

B. Related Works

Video summarization is gaining widespread attention in recent years. As mentioned in [1], many existing works focused on the problem of structured video summarization, such as the movie or sports videos [15], [16]. The specific characteristics of these videos help to achieve good video summarization results. However, these methods are usually not easy to be extended to general video sequences.

In general, key frame based method is one of the most commonly used techniques in video summarization. Features such as gradient orientations [14], color features [17] and a combination of color and texture features [18] were used. Beyond purely visual information, additional audio data [19] or multi-data source [20] were also considered as important features to find the key frames of a video. Object level methods [21]–[23] have also been applied to remove irrelevant video frames, in which the relationships between objects were considered. The works [3], [24] used images as a prior to create semantically meaningful summaries. Change detection was also used in video summarization [25], in which a video was clustered based on a spatio-temporal slice model. In [26], the authors developed a methodology to create a sort of scene context by computing shot similarity and exploiting visual attention. In [12] and [27], new video segmenting methods were developed for summarizing videos. Besides these, egocentric video summarization methods have also been

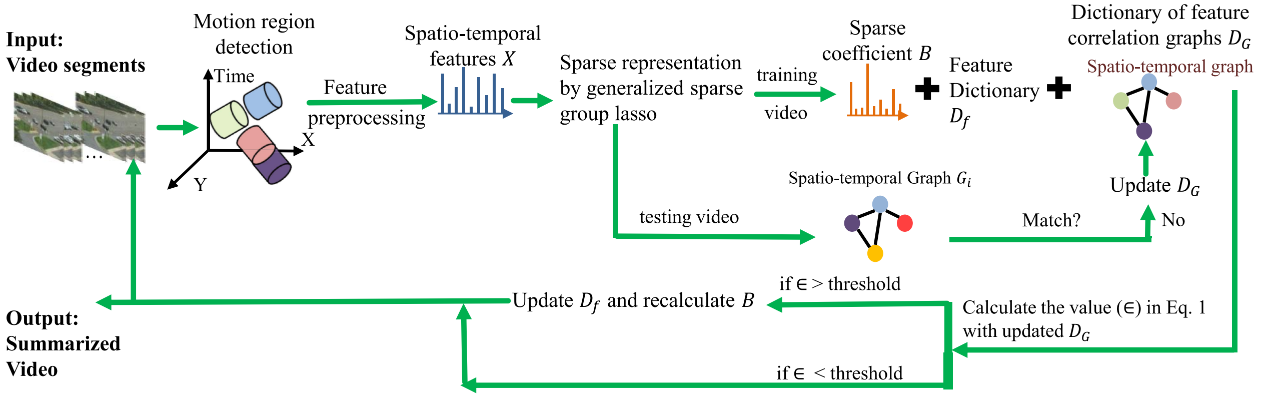

Input:
Video segments
Motion region detection
Time
X
Y
Feature preprocessing
Spatio-temporal features X
Sparse representation by generalized sparse group lasso
training video
Sparse coefficient B
Feature Dictionary D_f
Dictionary of feature correlation graphs D_G
Spatio-temporal graph
testing video
Spatio-temporal Graph G_i
Match?
Update D_G
No
if ∈ > threshold
Update D_f and recalculate B
Calculate the value (∈) in Eq. 1 with updated D_G
if ∈ < threshold
Output:
Summarized Video

Fig. 2. Overview of CAVS. ϵ represents the objective function in Eq. 1.

developed. In [6], a saliency based framework learned a linear regression model to predict importance score for each frame. However, these methods require special features and are event specific. So the applications are usually limited to the domain of wearable cameras.

Context information has been considered in video summarization. In [28], a random-walk based approach between video subshots was developed to indicate the progression of the events. In [29], the authors built the structure of the story according to the characters. the things, the places, and the time. In [30], the authors proposed a method to summarize the content of video search by mining and threading key shots, but the video tags were assigned to the whole videos instead of specific shots. In [31], the authors assigned the key-shot tags in a propagation process. However, all these methods did not fully investigate the high-level feature correlations in a short time period. Moreover, these methods cannot online update the features and their correlations, which is different from our approach.

Sparse coding, which has proved successful in problems of image classification [32], has been applied to the problem of video summarization in recent years. In [2], the features of the entire video were learned as a dictionary to reconstruct every video segment. Following [33], the method [7] improved the sparse coding method used in [2] by online updating of the learned dictionary. However, our method is significantly different from these approaches, where every feature vector was considered independently. In our model, the spatio-temporal dependencies between the features are incorporated into the sparse coding based video summarization framework. This is a more accurate representation of the actual video since it models the inter-relationships between the various objects and events. The works [5], [34] have explored the correlations between activities. However, the abnormal event detection method [34] considered the co-occurrence between pixels rather than events as in our approach and do not explicitly model the sparsity. The activity recognition work [5] required the prior knowledge of all the available activities in the dataset.

II. Video Summarization Methodology

An overview of the framework is shown in Fig. 2. A sparse coding model is built to learn a feature dictionary and a sparse representation of the video features. A dictionary of feature correlation graphs is obtained by learning the spatio-temporal correlations between video features. Given new video segments, if the video features in these segments can be sparsely represented by the learned features, the corresponding video segments are not important to summarize. Otherwise, the corresponding video segments are absorbed into the summarized video. We now describe a detailed overview of our proposed context-aware video summarization framework.

A. Feature Representation

Given a set of videos $\mathbf{Y}$, we use an adaptive background subtraction algorithm [35] to locate motion regions. Then, we evenly segment $\mathbf{Y}$ into small video segments $\{Y_1, Y_2, \cdots\}$. In every video segment Y_i, we use the spatio-temporal interest point (STIP) detector in [36] to generate concatenated histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features for the detected motion regions. A video segment, enriched with multiple events, is represented by histograms of STIP features. Note that other features like SIFT can also be used in our framework.

B. Problem Formulation

Sparse coding can find a set of basis vectors, i.e. the dictionary of the input feature matrix and the sparse coordinates with respect to the dictionary. In the training videos, our goal is to learn a feature dictionary D_f of most discriminative features that represents the whole feature set $X = \{X_1, X_2, \cdots\}$ of size $|\mathcal{X}|$, where $|\mathcal{X}|$ is the number of feature vectors in the training videos. The size of feature dictionary is denoted by $|\mathcal{D}_f|$. The dictionary of feature correlation graphs is denoted by D_g, the size of which is $|\mathcal{D}_g|$. Given the testing videos, we find a coefficient matrix B that minimizes the difference between the features in the training videos and those in the testing videos. We use $B_i = \{B_i^1, B_i^2, \cdots\} \in \mathbb{R}^{|\mathcal{D}_f|}$ to represent the i-th column of B, and B_i^j to represent the j-th item of B_i. Thus the video summarization problem can be formulated as

$$\min_B \frac{1}{2|\mathcal{X}|}\sum_{p=1}^{|\mathcal{D}_g|}\Big\{\left\|X - D_f B\right\|_F^2 + \alpha_1 \mathrm{Tr}(BL_pB^T) + \alpha_2\sum_{j=1}^{|\mathcal{D}_f|}\left\|B^j\right\|_2 + \alpha_3\sum_{i=1}^{|\mathcal{X}|}\|B_i\|_1\Big\}, \qquad (1)$$

where α_1, α_2 and α_3 are regularization parameters, $\|.\|_F$ denotes the matrix Frobenius norm, Tr represents the trace of a matrix, and L_p is a Laplacian matrix that is explained in details below. The first term in Eq. 1 indicates the reconstruction error, and the last two terms denote the group sparsity regularization. With the optimal B, Eq. 1 outputs the difference between the features in the new video segments and those in the existing videos, which is shown in Fig. 2. Ideally, if features in a video segment have not been observed, the reconstruction cost should be high and contain a large number of atoms in the dictionary.

There are two major contributions in Eq. 1 that makes our framework different from the existing works [2], [7] on video summarization. The first difference is the sparsity-inducing regularization term, which has been studied in the statistics and machine learning [37], [38]. It is often defined as the problem of group lasso. We, however, adopt the state-of-the-art methodology, the generalized sparse group lasso [8] to solve our problem. In the traditional sparse representation algorithms, l_1 norm is mostly used [39] and $l_{2,1}$ norm [2] is proved to perform better than l_1 norm in some cases. Recently, the study of group lasso has attracted more attention [37], [38]. It works like the lasso at the group level: the model can either keep or drop an entire group. However, the group lasso does not yield sparsity within a group. The advantage of the application of sparse group lasso over traditional l_1 norm, l_2 norm and group lasso is that it can find both “groupwise sparsity” and “within group sparsity”. Specifically, “groupwise sparsity” refers to the number of groups with at least one nonzero coefficient, and “within group sparsity” refers to the number of nonzero coefficients within every nonzero group.

Moreover, we introduce the term $\mathrm{Tr}(BL_pB^T)$ in Eq. 1 as a regularization terms introduced by the spatio-temporal correlations between features. The idea is inspired from [40] and [41], in which the dependencies between features are considered as a regularization term in the energy function. For the set of video segments p, we develop an undirected weighted graph $\mathcal{G}_p$ that models the spatio-temporal correlations between features. A dictionary of the graphs $D_g = \{\mathcal{G}_0, \mathcal{G}_1, \cdots, \mathcal{G}_p, \cdots\}$ denotes all the correlation graphs. In CAVS, M_p represents the spatio-temporal correlations between features, and thus makes CAVS summarize the global discriminative video portions. We define the degree matrix R_p as a diagonal matrix with each diagonal element as $\sum_k M_p^{ik}$, where M_p^{ik} is the element of the i-th row and the k-th column of M_p. $L_p = R_p - M_p$ is the Laplacian matrix. Given B which is a sparse representation of the feature matrix X, $\mathrm{Tr}(BL_pB^T)$ essentially represents how closely two feature vectors are correlated, and equals to $\frac{1}{2}\sum_i\sum_k (B_i - B_k)^T(B_i - B_k)M_p^{ik}$.

III. Optimization Methodology

In this section, we propose a methodology to solve Eq. 1. This includes the dictionary learning and updating processes.

A. Sparse Matrix Optimization

Eq. 1 is the summation of convex functions and is therefore convex. If we consider every column of X and B, the term $X_i - D_f B_i$ is a column vector, the length of which is the number of rows in X. It can be shown that Eq. 1 can be rewritten as

$$\min_B \frac{1}{2|\mathcal{X}|}\sum_{p=1}^{|\mathcal{D}_g|}\left\{\sum_{i=1}^{|\mathcal{X}|}\left\|X_i - D_f B_i\right\|_2^2 + \alpha_1\sum_{i,k=1}^{|\mathcal{X}|} L_p^{ik}B_i^T B_k + \alpha_2\sum_{j=1}^{|\mathcal{D}_f|}\left\|B^j\right\|_2 + \alpha_3\sum_{i=1}^{|\mathcal{X}|}\|B_i\|_1\right\}. \quad (2)$$

To find an optimal solution of B, we use the block coordinate based methodology that is able to yield sparse solutions at both the group and individual feature levels [8]. The solution can formulate the original problem in Eq. 2 to a convex function and a separable penalty. It has been proved that the group lasso criteria is separable, so the block coordinate descent can be used to optimize it. For each group, the coefficient’s feature-sign is used to determine the gradient of each group’s cost function. The problem is then reduced to a quadratic optimization problem. An overview of the optimization process is provided in Algorithm 1, and the details can be found in [8].

B. Learning Dictionary of Features and Feature Correlation Graphs

We adopt the method in [42] to learn the feature dictionary. A summary of the method is as follows.

1) CAVS generates a random dictionary with a fixed number of atoms.

2) Given the initial dictionary, the algorithm seeks a solution of the reconstruction matrix B.

3) The two step iteration process between parameters B and D continues until convergence. The updating process converges when the difference between the dictionary updating cost function with the previous B and that with the reconstructed B is smaller than a threshold. Please refer to [42] for more details.

In the process of learning the correlation matrix M_p, a function of L_p in Eq. 2, we build a spatio-temporal graph between features $\mathcal{G}_0 = (\mathbf{V}, \mathbf{E})$. The set of nodes is $\mathbf{V} = \{V_1, V_2, \cdots\}$ and the set of edges is $\mathbf{E} = \{\cdots, E_{ij}, \cdots\}$. Such a graph $\mathcal{G}_0$ is called a feature correlation graph. We use the method similar to [43]. Firstly, we use Bag-Of-Words combined with multi-class support vector machine (BOW+SVM) to calculate the posterior probability that a feature vector belongs to an activity class $p(c_j|x_i)$, where c_j denotes class j.[1] The node label V_i is $\arg\max_j p(c_j|x_i)$. The edge E_{ij} represents the spatio-temporal correlations between nodes V_i and V_j. The spatial correlation models the probability of a feature vector belonging to a particular class given its spatial distance with its neighbor. The temporal correlation models the probability of a feature vector belonging to a particular class given its temporal distance

[1] Note that the supervised SVM can be easily replaced by an unsupervised approach like nearest neighbor. Using a supervised approach is not a strong assumption because most application domains will have a set of commonly occurring activities which can be used to initialize the dictionary during a training phase.

Algorithm 1 Sparse Group Lasso Optimization

```
Input: Video features $X$
for $g \leftarrow 1$ to $|\mathcal{D}_g|$ do
    for $i \leftarrow 1$ to $|\mathcal{X}|$ do
        Define $\beta = B_i$;
        Initialize $\hat{\beta} = \beta_0$
        for $j \leftarrow 1$ to $|\mathcal{D}_f|$ do
            Define $H_i = X_i - \sum_{k \neq j} D_f^k \beta_k$,
            $D_f^j = (A_1, A_2, \cdots, A_k)$, where $D_f^j$ is the
            $j$-th group of $D_f$,
            $\beta_j = (\theta_1, \theta_2, \cdots, \theta_k)$,
            $\mathbf{v}_l = (w_1, \cdots, w_N) = H_i - \sum_{k \neq l} A_k \theta_k$.
            if $\theta_k \neq 0$ then
                $s_k = \theta_k / \beta_j$,
            else
                $s$ satisfies $\|s\|_2 \leq 1$
            end
            $t_k \in \text{sign}(\theta_k)$;
            $a_k = \alpha_2 s_k + \alpha_3 t_k$;
            $J(t) = \frac{1}{{\alpha_2}^2} \sum_l (a_l - \alpha_3 t_l)^2$ ;
            if $J(\hat{t}) \leq 1$ then
                Set $\hat{\beta}_j = 0$;
            else
                for All $l$ and $k$ do
                    if $|A_l^T \mathbf{v}_j| < \alpha_3$ then
                        $\hat{\theta}_l = 0$;
                    else
                        $\min_{\theta_l} \{ \frac{1}{2} \sum_{n=1}^{N} (v_n - \sum_{l=1}^{k} A_{nl} \theta_l)^2 +$
                        $\alpha_1 \sum_{n,l} L_{nl} \beta^T \beta + \alpha_2 \|\theta\|_2 +$
                        $\alpha_3 \sum_{l=1}^{k} |\theta_l| \}$ (*);
                    end
                end
            end
        end
    end
end
(*) can be solved by the method in [8];
Output: $B$;
```

with its neighbor. Given two nodes V_i and V_j and their spatial and temporal locations s and t, the spatial and temporal correlations ψ_s and ψ_t are modeled as normal distributions

$$\psi_s(V_i, V_j) = \mathcal{N}(\|s_i - s_j\|^2; \mu_s(c_i, c_j), \sigma_s(c_i, c_j)),$$
$$\psi_t(V_i, V_j) = \mathcal{N}(\|t_i - t_j\|^2; \mu_t(c_i, c_j), \sigma_t(c_i, c_j)), \quad (3)$$

where $\mu_s(c_i, c_j), \sigma_s(c_i, c_j)$ are the parameters of the spatial correlation and $\mu_t(c_i, c_j), \sigma_t(c_i, c_j)$ are the parameters of the temporal correlation. The edge weight, represented by spatio-temporal correlation between two nodes V_i and V_j, is calculated by

$$\Psi_{ij} = u_{ij} \psi_s(V_i, V_j) \psi_t(V_i, V_j), \quad (4)$$

where u_{ij} is an association probability that is computed as a ratio of the number of times a feature class c_j has occurred in the vicinity of c_i to the total number of times c_i has occurred. The vicinity represents the spatio-temporal relationships between two activities. If the activities are spatially or temporally close to each other, e.g., the distance is less than a threshold, one activity is in the vicinity of another one. The parameters can be learned by maximizing $\sum_k \Psi_{ij}^k$, where Ψ_{ij}^k is the k-th training example. This can be done by the maximum likelihood algorithm. The typical values of the parameters are described in Sec. IV. We assume that every edge weight can be learned independently. An example of a learned graph is shown in Fig. 3.

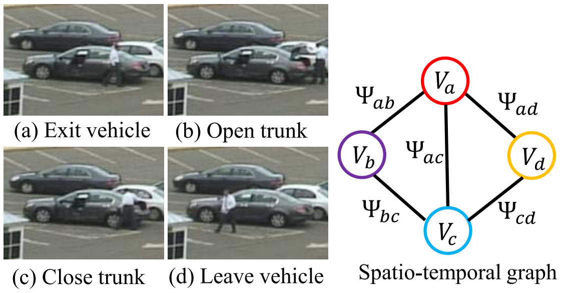

(a) Exit vehicle (b) Open trunk (c) Close trunk (d) Leave vehicle Spatio-temporal graph

Fig. 3. An example of spatio-temporal graph learning. (a)-(d) show four events which are correlated to each other. The spatio-temporal graph is learned by the correlations between these events.

Our algorithm scans through consecutive video segments in the training dataset and models the pairwise spatio-temporal correlations between every pair of feature vectors. In the training of CAVS, the dictionary of feature correlation graphs is initialized as $D_G = \{\mathcal{G}_0\}$, which is built by the method above. Each item of M is represented by the correlation graph $\mathcal{G}_0$. Specifically, given that $c_m = i$ and $c_n = j$, the correlation between two features X_m and X_n is $M_{mn} = \Psi_{ij}$.

C. Online Dictionary Update of Features and Feature Correlation Graphs

As CAVS scans through the video sequence, features that cannot be sparsely reconstructed using the existing dictionary, are considered to belong to video segments that are parts of the summarized video. Video watchers do not want to watch similar video portions again. Therefore, updating the dictionaries is of great importance.

In the process of online updating the dictionary D_f, we follow the method in [42]. Concretely, CAVS updates the dictionary sequentially, and only needs to store two matrices: $P_t = \sum_{i=1}^{t} B_i B_i^T$ and $Q_t = \sum_{i=1}^{t} X_i^T B_i^T$. Given D_f at time $t-1$, we use the sparse coding steps to compute B at time $t-1$. With these two variables at time $t-1$, the algorithm can find the new optimal D_f at time t, where each column of D_f is updated sequentially. It has been proved in [42] that the dictionary D_f at time $t-1$ is a warm restart for computing D_f at time t, and this process can converge to an optimal solution.

When updating the dictionary of feature correlation graphs, new feature correlation graphs are constructed by the method in Sec. III-B. This process is performed independent of the learning process. If the new graph is recognized as different from the graphs in the dictionary, the new graph is incorporated into the graph dictionary. The methodology in [44] is used to compare the similarity between the built graph and the learned graphs in the dictionary. We calculate the similarity between the new graph $\mathcal{G}_i$ and the graphs in the dictionary $\mathcal{D}_\mathcal{G}$. The similarity between $\mathcal{G}_i$ and a graph in the dictionary $\mathcal{D}_{\mathcal{G}_l}$ is denoted by $sim(\mathcal{G}_i, \mathcal{D}_{\mathcal{G}_l})$. Assume that there are two graphs $\mathcal{G}_a = (\mathbf{V}^a, \mathbf{E}^a)$ and $\mathcal{G}_b = (\mathbf{V}^b, \mathbf{E}^b)$. The number of nodes in these two graphs are represented by $|\mathcal{V}^a|$ and $|\mathcal{V}^b|$ individually. A solution of graph matching is a subset of possible correspondences, denoted by a binary matrix H with the size $|\mathcal{V}^a| \times |\mathcal{V}^b|$. If $V_i^a \in \mathbf{V}_a$ matches $V_{i'}^b \in \mathbf{V}_b$, then $H_{ii'} = 1$; otherwise $H_{ii'} = 0$. We use h to represent a columnwise vectorized replica of H. The graph matching problems can be defined as the problem of finding the assignment vector h^* that satisfies

$$h^* = \arg\max_h sim(\mathcal{G}_a, \mathcal{G}_b)$$
$$s.t. \begin{cases} h \in \{0,1\}^{|\mathcal{V}^a||\mathcal{V}^b|}, \\ \sum_i h_{ii'} \leq 1, \quad \sum_{i'} h_{ii'} \leq 1. \end{cases} \tag{5}$$

The similarity function $sim(\mathcal{G}_a, \mathcal{G}_b)$ is decomposed into the node similarity function $s_v(i, i')$ for a node pair $V_i \in \mathbf{V}_a$ and $V_{i'} \in \mathbf{V}_b$, and an edge similarity function $s_e(ij, i'j')$ for an edge pair $E_{ij} \in \mathbf{E}_a$ and $E_{i'j'} \in \mathbf{E}_b$. The similarity function is thus defined as

$$sim(\mathcal{G}_a, \mathcal{G}_b) = \sum_{h_{ii'}=1} s_v(i, i') + \sum_{h_{ii'}=1, h_{jj'}=1} s_e(ij, i'j'), \tag{6}$$

where $s_v(i, i')$ is 1 if the distance between the features of V_i and $V_{i'}$ is smaller than a threshold, and 0 otherwise. $s_e(ij, i'j')$ is 1 if the difference between the weights on two edges is smaller than a threshold, and 0 otherwise.

Note that updating graph correlation is an unsupervised process, where prior information of class labels is not needed. When some activities are detected, they are compared with the known ones based on the individual features in each detected region, as well as the inter-relationships between them. This is done by comparing with the nodes and edges of the learned graphical model as available up that time. If the individual activities and their inter-relationships do not match, they are identified as new ones, and the graph is updated. We update the dictionary of correlation graphs based on Algorithm 2, which is similar to that in [45] and [46]. The idea is to calculate the similarity between two graphs. If it is higher than a threshold, the edges of the graph are updated accordingly.

As shown in [45], the computation cost of matching two graphs is $O(|\mathcal{V}|^3)$, where $|\mathcal{V}|$ is the average number of nodes in the correlation graphs. Assuming W_1 to be the size of correlation graph dictionary, and W_2 to be the number of testing graphs. Every testing graph is matched to every graph in the correlation graph dictionary, which has W_1 graphs at most. So, the overall computational cost of the proposed method is $O(W_1 W_2 |\mathcal{V}|^3)$. As we can see, the computational cost is linear in the size of correlation graph dictionary W_1 and the number of testing graphs W_2. In the experiments, we select a relative small number of time window, and therefore, the number of nodes $|\mathcal{V}|$ is usually small (typically less than 10). As a result, $|\mathcal{V}|^3$ is small in our problems and thus, the proposed method can be scaled to large scale datasets.

Algorithm 2 Online Update the Dictionary of Correlation Graphs

```
Input: The learned weight graph $\mathcal{G}_0$ from the training
       videos and the new graph $\mathcal{G}_i$ which is built from
       the new video segment, and a threshold $\tau$.
Initialization: Let $D_\mathcal{G} = \mathcal{G}_0$;
for $l \leftarrow 1$ to $|\mathcal{D}_\mathcal{G}|$ (the size of $\mathcal{D}_\mathcal{G}$) do
    if $sim(\mathcal{G}_i, D_{\mathcal{G}_l}) > \tau$ then
        for $j \leftarrow 1$ to $|E_{\mathcal{G}_i}|$ (the size of edges in $\mathcal{G}_i$) do
            for $j' \leftarrow 1$ to $|E_{D_{\mathcal{G}_l}}|$ (the size of edges in
            $D_{\mathcal{G}_l}$) do
                Denote the nodes associated with the $j$-th
                edge in $\mathcal{G}_i$ as $V_p$ and $V_q$, and those
                associated with the $j'$-th edge in $D_{\mathcal{G}_l}$ as
                $V_{p'}$ and $V_{q'}$ ;
                if $s_v(p, p') = 1$ and $s_v(q, q') = 1$ then
                    if $s_e(j, j') = 1$ then
                        Accept the original edge weight
                        between $j$ and $j'$;
                    else
                        $E_{p'q'}^{\mathcal{D}_{\mathcal{G}_l}} \leftarrow E_{pq}^{\mathcal{G}_i}$;
                    end
                end
            end
        end
    else
        $\mathcal{D}_{\mathcal{G}_l} = \mathcal{D}_{\mathcal{G}_l} \cup \mathcal{G}_i$;
    end
end
Output: $\mathcal{D}_\mathcal{G}$;
```

IV. Experiments

To show the effectiveness of CAVS, we perform experiments on four public datasets: UCLA office dataset [11], VIRAT dataset [10], SumMe dataset [12] and TVSum50 Dataset [13]. All datasets consist of various videos and contain many different events. The strength of our proposed approach can be seen in surveillance videos (first two datasets). Surveillance videos have a lot of redundancy and, hence, there is a need to summarize them. Such videos exhibit strong spatio-temporal relationships between activities and hence these should be considered in the summarization process. In addition, we worked on the last two datasets, where some user-generated video sequences are selected to test the broad effectiveness of the proposed approach. We compare the results of CAVS with the state-of-the-art methods.

A. Dataset

The UCLA office dataset consists of three surveillance videos of single and two-person activities. The total length of these three video sequences is around 35 minutes. Every video sequence is composed of repetitive events with different temporal orders. We use one third of every video sequence to

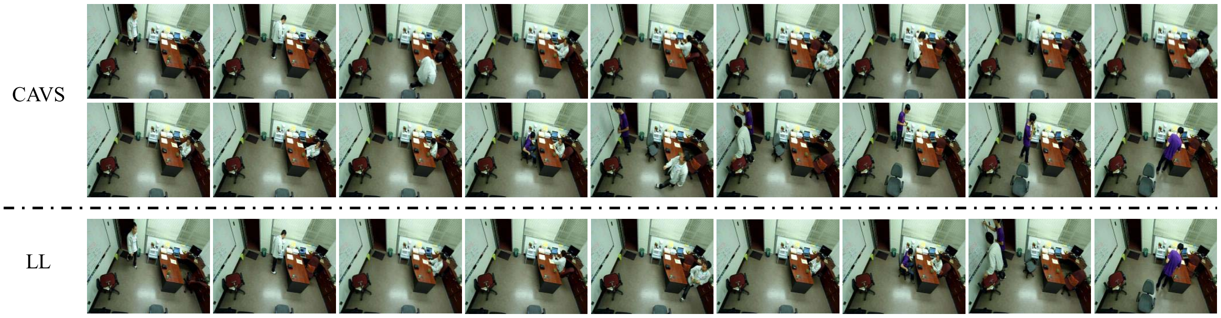


Fig. 4. Video summarization results on UCLA office dataset. The results by CAVS are a series of stories, while LL obtains the results that are purely based on the independent video features. In the results of CAVS, the first 12 figures represent stories of temporal events. Then the spatial correlated events are captured. The supplemental material provides the videos for clear video summaries.

train our model, and use the rest two thirds to online update the dictionaries, thus producing the summaries.

The VIRAT dataset is a surveillance dataset which contains many challenging characteristics such as large variation in the activities and clutter in the scene. Moreover, there are many different spatio-temporal correlations between events that make VIRAT dataset more challenging than other datasets used in the existing video summarization works. A surveillance dataset usually does not have a specific topic; thus most summarization algorithms working on storyline based videos [3], [27] cannot be directly applied. In VIRAT, there are 334 videos, each lasting 2 to 15 minutes. These videos are recorded on 10 different scenarios including parking lots, university campuses and etc. We use around 40% of the dataset as training and the rest as testing.

The SumMe dataset consists of videos from both static and moving cameras. Every video sequence lasts from 30 secs to 7 mins. The topics of SumMe dataset cover holidays, events and sports. We select nine video sequences in which enough person/animal activity features can be extracted to test CAVS algorithm. Similar to UCLA office dataset, one third of every video sequence is used to train the model. The rest two thirds of a video sequence are used to update the dictionaries. The TVSum50 dataset contains 50 videos which are collected from YouTube. The topics of these videos include news, interviews, documentaries, and user-generated content such as egocentric. We select four videos to show the effectiveness of our method. In these videos, we use 10 % as the training data.

B. Results

To find a compact representation of the activity features, we use the spatio-temporal pyramid and average pooling method to generate a vector of size 162 (HoG+HoF) features. In CAVS, we fix the number of atoms in the dictionary to be 120. Three parameters in Eq. 2 are manually set to be: $\alpha_1 = 0.3$, $\alpha_2 = 0.05$ and $\alpha_3 = 0.08$. The length of every video segment is set to be 90 frames (30 frames per second). The typical values in Eq. 3 are $\mu_s = 0.1*\text{VidwoframeWidth}$, $\sigma_s = 0.1*\text{VidwoframeWidth}$, $\mu_t = 90\text{frames}$, and $\sigma_t = 30\text{frames}$.

We adopt two evaluation metrics on different datasets that are used in this paper. We use the evaluation metrics in [2] and [7] on VIRAT, UCLA and TVSum50 datasets because these two works are mostly closely related to our work and we directly compare our results with theirs on these datasets. The summarization accuracy is reported by this evaluation method, in which both video segment contents and time differences are considered in this evaluation method. Specifically, if two video segments share the same scene contents and occur within a period of time, they are considered to be equivalent to each other. The ground truth summary is manually labeled by two analysts to minimize the influence of subjectiveness. In the evaluation process, the summarization accuracy is computed as the ratio between the automatically summarized video and the ground truth summary provided by two analysts.

TABLE I

Video Summarization Results on UCLA Office Dataset. "Time" Represents the Total Length of the Original Videos. The Percentage Value Represent the Overlaps Between the Summarized Video and the Ground Truth

	Time(s)	AC	DSVS	LL	CAVS
UCLA 1	420	67.9%	75.3%	83.0%	88.5%
UCLA 2	324	71.2%	72.5%	73.2%	76.7%
UCLA 3	1154	58.5%	66.6%	69.5%	78.2%
Average	-	65.9%	71.5%	75.2%	81.3%

Moreover, we adopt the evaluation methodology in [12] to compare our results with theirs on SumMe dataset since this work directly reported their results on SumMe dataset. Specifically, the evaluation score F_i for the human selection i is defined as

$$F_i = \frac{1}{N-1} \sum_{j \neq i} 2 \frac{p_{ij} r_{ij}}{p_{ij} + r_{ij}} \tag{7}$$

where N is the number of humans, p_{ij} is the precision and r_{ij} is the recall of human selection i using the j-th ground truth.

Table I illustrates the summarization accuracy on UCLA dataset with different algorithms. We compare our algorithm with activity clustering video abstraction (AC) [14], dictionary selection based video summarization (DSVS) [2] and LiveLight (LL) [7]. It is shown that CAVS performs the best among all the three scenarios. An illustration of the results on UCLA dataset can be found in Fig. 4, where selective pictorial results of CAVS and LL are shown individually. CAVS generates a summarized video which is composed of short stories. Although some events are summarized more

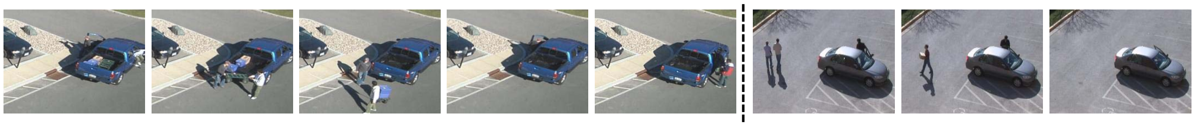


Fig. 5. Representative video summarization results on VIRAT by CAVS. These two stories are not summarized by the other methods, because every single event is a repeat of the events in the training videos. However, the stories are captured by our algorithm through the spatio-temporal correlations between events. The supplemental material provides the videos for clear video summaries.

TABLE II

VIDEO SUMMARIZATION RESULTS ON VIRAT DATASET. "TIME" REPRESENTS THE TOTAL LENGTH OF THE ORIGINAL VIDEOS. THE PERCENTAGE VALUES REPRESENT THE OVERLAPS BETWEEN THE SUMMARIZED VIDEO AND THE GROUND TRUTH

	Time(s)	AC	DSVS	LL	CAVS
VIR 1	1880	48.0%	67.3%	68.0%	77.1%
VIR 2	986	52.5%	66.5%	76.2%	76.8%
VIR 3	1656	50.4%	69.0%	74.1%	83.2%
VIR 4	1441	60.2%	65.5%	64.3%	75.1%
VIR 5	942	58.5%	64.0%	64.7%	68.7%
VIR 6	2052	60.5%	71.5%	71.6%	71.0%
VIR 7	675	59.6%	72.9%	73.3%	83.4%
VIR 8	305	79.9%	85.6%	90.0%	90.0%
VIR 9	1546	61.3%	74.5%	78.0%	82.2%
VIR 10	631	81.6%	89.7%	90.7%	92.8%
Average	-	61.2%	72.7%	75.0%	80.0%

TABLE III

VIDEO SUMMARIZATION RESULTS ON TVSUM50 DATASET. "TIME" REPRESENTS THE TOTAL LENGTH OF THE ORIGINAL VIDEOS. THE PERCENTAGE VALUE REPRESENT THE OVERLAPS BETWEEN THE SUMMARIZED VIDEO AND THE GROUND TRUTH

	Time(s)	AC	DSVS	LL	CAVS
Video 1	405	56.0%	65.2%	64.1%	65.0%
Video 2	397	46.5%	55.5%	57.2%	57.7%
Video 3	104	56.5%	63.6%	63.5%	63.5%
Video 4	154	47.0%	54.4%	55.5%	58.0%
Average	-	51.5%	59.7%	60.1%	61.0%

TABLE IV

VIDEO SUMMARIZATION RESULTS ON SUMME DATASET. "TIME" REPRESENTS THE TOTAL LENGTH OF THE ORIGINAL VIDEOS. WE USE THE PRECISION MEASURE IN EQ. 7 FOR BOTH SF AND CAVS

	Time(s)	SF	CAVS
Bearpark	133	0.12	0.38
Bike Polo	103	0.36	0.30
Cooking	86	0.32	0.47
Excavators	388	0.19	0.32
Jumps	39	0.43	0.34
Kids Playing	106	0.09	0.40
Paluma jump	85	0.18	0.29
Playing water	102	0.20	0.42
Saving dolphins	222	0.15	0.24
Average	-	0.23	0.35

than once, the spatio-temporal correlations between them tell analysts a whole story of what happens in the video. For instance, a short story is composed of a person working on the laptop, standing up, pouring water and sitting down. However, LL only summarizes the events of working on a laptop and pouring water, which are not informative to analysts. Similarly, another story could be a person pouring water, picking up a phone and placing down a phone. Such strong correlations are not detected by LL.

Table II shows a summary of the results on VIRAT dataset. In VIRAT, we classify the videos into 10 categories based on the type of scenarios. It can be seen that CAVS obtains the best results in most scenarios on the average accuracy in VIRAT. In scenario 8, LL and CAVS obtain the same results. This is because of the few spatio-temporal feature correlations in this scenario.

Fig. 5 shows two examples from the summarized videos in VIRAT by CAVS. We select some key frames from the highlighted video segments to represent two stories that CAVS summarizes. The first story is that two persons get off the truck, load objects, one leaves and one goes back into the truck, and a person loads objects into the truck. The second story shows the story that a person gets out of the vehicle, another person loads an object while the first person opens the door, and the second person leaves and the first person goes back into the car. With the method of [2] and [7], the video features in these video segments can be sparsely represented by those in the training videos, and these scenes are not summarized. CAVS, however, identifies these in the summarized video.

We also extend our approach to non-surveillance videos in order to show its usage on other type of videos. Table III shows a summary of the results on TVSum50 dataset. We select four representative videos from this dataset. Note that CAVS achieved the highest average accuracy. The difference with other methods is smaller because this dataset has less inter-activity correlations than the surveillance datasets considered earlier.

In Table IV, we illustrate our results on SumMe dataset with the evaluation metrics used in [12], where SF denotes the superframe method in [12]. It is shown that CAVS obtain significant better results than SF (the superframe method in [12]. 9 video sequences with rich human/animal activity features are tested by our algorithm. Some representative image results on the Kids Playing video sequence and the Cooking video sequence are shown in Fig. 6, and a representative scenarios in TVSum50 dataset is also shown. In the first row, we show the summarized kids activities which include lying on the leaves, picking up leaves, standing up, throwing leaves at others, running away, running back and throwing leaves again. These activities and their temporal correlations are well captured by CAVS. In the second row, the activities of cooking meats, moving onion slices, stacking up onion cones, adding oils, burning onions are well captured. In the last image, we can

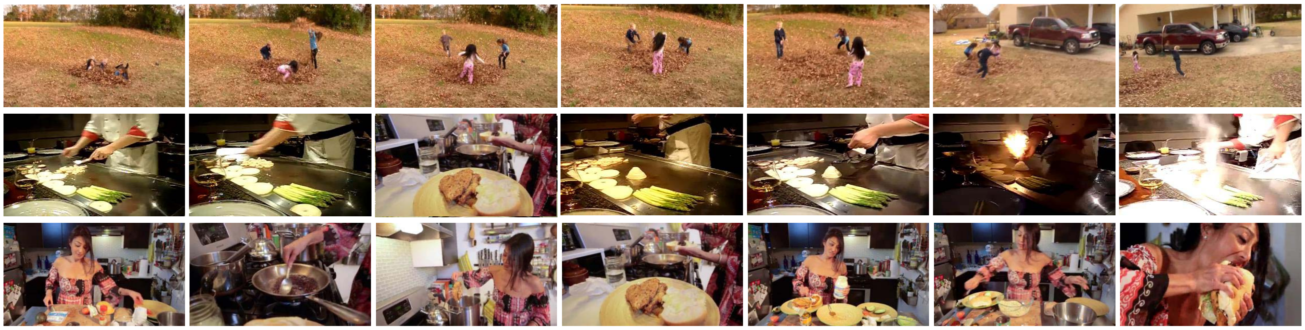


Fig. 6. Summarized videos in SumMe and TVSum50. The activities of lying on the leaves, picking up leaves, standing up, throwing leaves to others, running away, running back and throwing leaves again are highlighted in the first row. The activities of cooking meats, moving onion slices, stacking up onion cones, adding oils, burning onions are well captured in the second row. In the third row, the important activities such as whisking eggs, frying meats, pouring sauces and eating foods are shown.

see that the spatial correlations between foods and fires are captured. In the third row, the important activities such as whisking eggs, frying meats, pouring sauces and eating foods are captured.

Our summarized video provides a 7x-40x compression without losing the semantic understanding of the original surveillance video. For instance, the lengths of CAVS summarized videos in UCLA dataset are 39s, 18s and 68s respectively; while those of LL summarized videos are 24s, 12s and 38s respectively. Although the CAVS summarized video is longer than that in [7], it is more informative and tells a whole story of how events interact with each other.

C. Discussion

Since our framework captures the features as well as their spatio-temporal correlations, the summarized outputs maybe longer, but more meaningful, than existing frameworks. Therefore, there may be some redundancy in the summarized output. There are two ways to reduce such redundant information. First, the proposed CAVS method can work with any lower-level video summarization methods, and further compress the selected segments. Moreover, we can control the amount of redundant information by changing the threshold that is explained in Fig. 2, which is the approach that we adopt in this paper. In the datasets of UCLA and VIRAT, which are surveillance datasets, we use a smaller threshold, compared to user-generated videos, which can increase the length of the summary. The reason is that for surveillance videos it is important not to miss anything important in the summary. In our summarized video, similar activities can still be part of the summary if their relationships with other activities are different. However, in the other two datasets, which are mostly composed of user-generated videos, e.g. from social media and news, the contents have been edited and short snippets presented. They do not contain the long-term spatio-temporal correlations, and thus the threshold is set to be larger than that is used in the surveillance datasets.

Moreover, there could be an extreme case which is that all the test sequences' features are out of the dictionary (i.e., cannot be reconstructed by the existing dictionary). In our experiments, if a feature of the testing sequence is out of the dictionary, the online learning algorithm will update the dictionary to include the new feature. This updated dictionary can then be considered in future summaries.

V. Conclusion

In this paper, we present a novel approach to summarize the most informative video portions. The main goal of this paper is to summarize the surveillance videos, but we have also shown its performance on a small number of other user-generated videos. Both individual local motion regions and interactions between these motion regions are taken into consideration in our framework. We formulate the video summarization problem as the problem of sparse feature reconstruction with generalized sparse group lasso. To solve the overall problem, we propose an algorithm to learn and update dictionaries of video features along with feature correlations. Our promising experimental results on two public datasets have shown that encapsulating the spatio-temporal correlations between events can be used to tell analysts a story of global events. Such a summarized video is closer to the ground truth than existing works.

References

[1] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 3, no. 1, pp. 1–37, 2007.

[2] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 66–75, Feb. 2012.

[3] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of Web images and videos for storyline reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4225–4232.

[4] R. Mottaghi *et al.*, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 891–898.

[5] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2491–2498.

[6] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1346–1353.

[7] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2513–2520.

[8] J. Friedman, T. Hastie, and R. Tibshirani. (2010). "A note on the group lasso and a sparse group lasso." [Online]. Available: http://arXiv:1001.0736

[9] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Statist.*, vol. 22, no. 2, pp. 231–245, 2013.

[10] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3153–3160.

[11] M. Pei, Y. Jia, and S.-C. Zhu, “Parsing video events with goal inference and intent prediction,” in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 487–494.

[12] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 505–520.

[13] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing Web videos using titles,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5179–5187.

[14] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg, “Clustered synopsis of surveillance video,” in *Proc. Int. Conf. Adv. Video Signal Surveill.*, 2009, pp. 195–200.

[15] C.-C. Cheng and C.-T. Hsu, “Fusion of audio and motion information on HMM-based highlight extraction for baseball games,” *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 585–599, Jun. 2006.

[16] B. Li and I. Sezan, “Semantic sports video analysis: Approaches and new applications,” in *Proc. Int. Conf. Image Process.*, 2013, pp. I-17–I-20.

[17] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “Summarizing visual data using bidirectional similarity,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[18] X. Zhu, X. Wu, J. Fan, A. K. Elmagarmid, and W. G. Aref, “Exploring video content structure for hierarchical summarization,” *Multimedia Syst.*, vol. 10, no. 2, pp. 98–115, 2004.

[19] W. Jiang, C. Cotton, and A. C. Loui, “Automatic consumer video summarization by audio and visual analysis,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011, pp. 1–6.

[20] X. Zhu, C. C. Loy, and S. Gong, “Video synopsis by heterogeneous multi-source correlation,” in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 81–88.

[21] S. Feng, Z. Lei, D. Yi, and S. Z. Li, “Online content-aware video condensation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2082–2087.

[22] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, “Webcam synopsis: Peeking around the world,” in *Proc. Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[23] S. Zhang and A. K. Roy-Chowdhury, “Video summarization through change detection in a non-overlapping camera network,” in *Proc. IEEE Conf. Image Process.*, Sep. 2015, pp. 3832–3836.

[24] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, “Large-scale video summarization using Web-image priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2698–2705.

[25] A. Hanjalic and H. Zhang, “An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1280–1289, Dec. 1999.

[26] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, “Video summarization and scene detection by graph modeling,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[27] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, “Category-specific video summarization,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 540–555.

[28] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2714–2721.

[29] B.-W. Chen, J.-C. Wang, and J.-F. Wang, “A novel video summarization based on mining the story-structure and semantic relations among concept entities,” *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 295–312, Feb. 2009.

[30] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua, “Beyond search: Event-driven summarization for Web videos,” *ACM Trans. Multimedia Comput.*, vol. 7, no. 4, 2011, Art. no. 35.

[31] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, “Event driven Web video summarization by tag localization and key-shot identification,” *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 975–985, Aug. 2012.

[32] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[33] B. Zhao, L. Fei-Fei, and E. P. Xing, “Online detection of unusual events in videos via dynamic sparse coding,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3313–3320.

[34] Y. Benezeth, P.-M. Jodoin, V. Saligrama, and C. Rosenberger, “Abnormal events detection based on spatio-temporal co-occurences,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2458–2465.

[35] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Proc. Int. Conf. Pattern Recognit.*, 2004, pp. 28–31.

[36] I. Laptev, “On space-time interest points,” *Int. J. Comput. Vis.*, vol. 64, no. 2, pp. 107–123, 2005.

[37] L. Meier, S. Van De Geer, and P. Bühlmann, “The group lasso for logistic regression,” *J. Roy. Statist. Soc. B*, vol. 70, no. 1, pp. 53–71, 2008.

[38] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *J. Roy. Statist. Soc. B*, vol. 68, no. 1, pp. 49–67, 2006.

[39] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in *Proc. 19th Ann. Conf. Neural Inf. Process. Syst.*, 2007, pp. 801–808.

[40] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14. 2001, pp. 585–591.

[41] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, “Graph regularized sparse coding for image representation,” *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

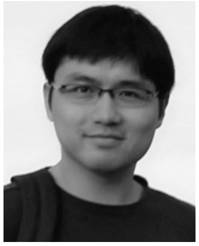
[42] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[43] B. Yao and L. Fei-Fei, “Modeling mutual context of object and human pose in human-object interaction activities,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 17–24.

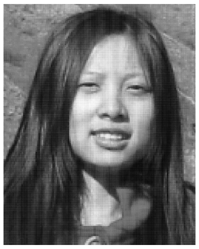
[44] F. Zhou and F. De la Torre, “Factorized graph matching,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 127–134.

[45] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *Proc. IEEE Conf. Comput. Vis.*, Oct. 2005, pp. 1482–1489.

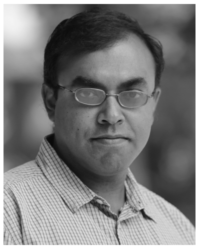
[46] U. Gaur, Y. Zhu, B. Song, and A. K. Roy-Chowdhury, “A ‘string of feature graphs’ model for recognition of complex activities in natural videos,” in *Proc. IEEE Conf. Comput. Vis.*, Nov. 2011, pp. 2595–2602.



Shu Zhang received the B.E. degree from Tianjin University, China, in 2007, the M.S. degree in electrical engineering from the University of Missouri, Columbia, in 2010, and the Ph.D. degree in electrical engineering from University of California at Riverside, Riverside, CA, USA, in 2015. He is currently a Senior Research Engineer with SONY Electronics Inc. His main research interests include computer vision, deep learning, and image processing.



Yingying Zhu received the M.S. degree in engineering from Shanghai Jiao Tong University in 2007, the M.S. degree in engineering from Washington State University, in 2010, and the Ph.D. degree from the Department of Electrical Engineering, University of California at Riverside, Riverside, CA, USA, in 2014. She is currently a Software Engineer with Google Inc. Her research interests include computer vision, pattern recognition and machine learning, image/video processing and communication.



Amit K. Roy-Chowdhury received the bachelor’s degree in electrical engineering from Jadavpur University, Calcutta, India, the master’s degree in systems science and automation from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park. He is a Professor of Electrical Engineering with the University of California at Riverside, Riverside, CA, USA. His research interests include image processing and analysis, computer vision, and video communications, and statistical methods for signal analysis. His current research projects include intelligent camera networks, wide-area scene analysis, motion analysis in video, activity recognition and search, video-based biometrics (face and gait), biological video analysis, and distributed video compression. He is co-author of the book entitled *The Acquisition and Analysis of Videos over Wide Areas*. He is an Editor of the book entitled *Distributed Video Sensor Networks*. He has been on the organizing and program committees of multiple conferences and serves on the editorial boards of a number of journals.