Laplacian Sparse Coding of Scenes for Video Classification

Yifang Yin
Interactive and Digital Media Institute
National University of Singapore
Singapore
Email: idmyiny@nus.edu.sg

Zhenguang Liu, Satyam, Roger Zimmermann
School of Computing
National University of Singapore
Singapore
Email: {liuzheng,satyam,rogerz}@comp.nus.edu.sg

***Abstract*—The challenging task of dynamic scene classification in unconstrained videos has drawn much research attention in recent years. Most existing work has focused on extracting local descriptors from spatiotemporal interesting points or subregions, followed by feature aggregation with advanced coding techniques. In this study, we analyse the effectiveness of global image descriptors and propose a novel Laplacian Sparse Coding of Scenes (LSCoS) method for video categorization. Previous methods neglect the semantic relationship among the visual scenes in the dictionary, resulting in generating different representations for videos with similar content. Intuitively, the coefficients assigned to the visual scenes of the same class should be promoted or demoted simultaneously for consistency concerns. To build upon the above ideas, we construct a Laplacian matrix by exploiting the connections between the representative scenes from each class and formulate the objective function with ℓ_1 and Laplacian regularizers to generate more robust semantically consistent sparse codes. Comprehensive experiments have been conducted on two public dynamic scene recognition datasets, namely Maryland and YUPENN. Experimental results demonstrate the effectiveness of our proposed approach, as our solution achieves the state-of-the-art classification rates and improves the accuracy by 2.86% $\sim$ 16.93% compared with the existing methods.**

***Keywords*-Video classification; dynamic scene understanding; Laplacian sparse coding; high-level video representation**

I. Introduction

Dynamic scene understanding has long been one of the fundamental challenges in computer vision. Here, "scene" refers to a place where an action or event occurs [1]. Over the past decade, extensive research has been carried out on representative feature extraction for scene understanding and classification [2], [3]. Descriptors of the spatiotemporal interesting points or subregions have been proposed, such as spatiotemporal oriented energies (SOE) [1], complementary spacetime orientation (CSO) [4], GIST3D [5], bags of spacetime energies (BoSE) [6], *etc.* The Bag-of-Visual-Words (BoVW) [7] model is usually adopted for feature aggregation and pooling. Though promising results have been reported, Penatti *et al.* [8] argued that local features might contain less semantic information than scene descriptors and therefore proposed a high-level video representation named Bag-of-Scene (BoS). However, as BoS was originally proposed for video geotagging, its effectiveness has not been validated for other video applications such as categorization and retrieval. Moreover, to the best of our knowledge, the existing feature coding techniques mostly neglect the semantic connections between the basis vectors in a codebook, which greatly hinders obtaining a better accuracy in video categorization [9].

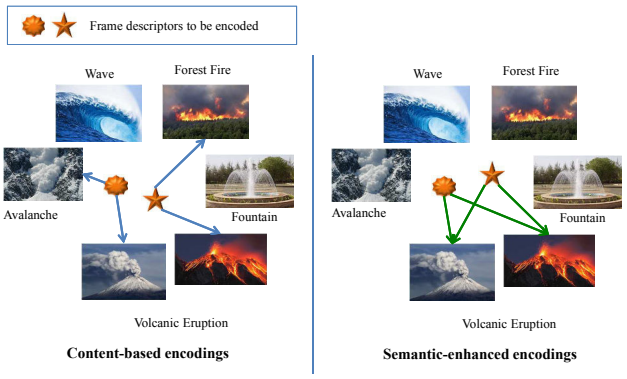


Figure 1. Comparison of the proposed semantic-enhanced Laplacian sparse coding of scenes and previous content-based feature quantization strategies.

To solve the aforementioned issues, we propose a novel Laplacian sparse coding technique for dynamic scene classification. A comparison of our proposed method and the previous visual similarity-based coding strategy is illustrated in Fig. 1. Assume that we have two pictures of volcanic eruption with different visual appearances to be encoded. Traditional approaches such as hard [10] or soft assignment [11] and sparse coding [12] assign features to one or more basis vectors in the dictionary purely based on the visual similarities. Due to the semantic gap, the visually similar neighbors of the two input volcanic eruption pictures may belong to different scene classes and therefore show little consistency in the semantic space (see the left of Fig. 1). It indicates that the codes generated for similar scenes may vary a lot without considering the mutual dependence among the pictures from the same class in the codebook. To overcome this drawback, we propose a Laplacian sparse coding of scenes technique which generates more robust codes by assigning coefficients to semantically similar neighbors (see the right of Fig. 1). Note that our work differs from Gao *et*

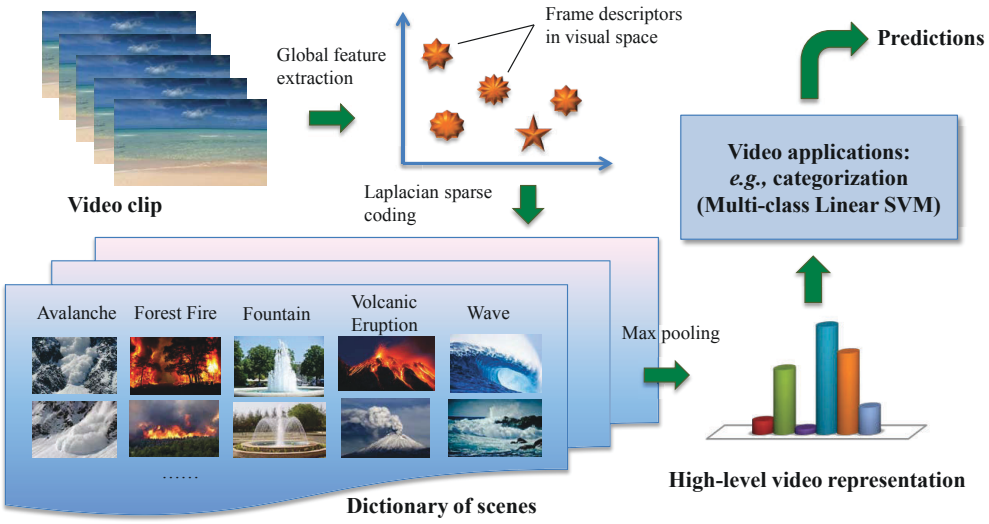


Figure 2. Overview of the proposed video feature encoding and classification framework.

al. [13] as they only improve the consistency of content-based feature quantization. Our method, on the other hand, emphasizes the semantic connections between the elements in the dictionary and therefore improves the consistency in the sparse codes of scenes with similar semantics.

The overview of our video classification framework is illustrated in Fig. 2. We process a video as a sequence of frames where each frame is initially represented by low-level visual features such as color [14], texture [15], and HOG [16]. To perform feature aggregation, we build a dictionary to encode the frame descriptors by selecting a set of representative pictures from different classes. For instance, Fig. 2 illustrates a dictionary of scenes including avalanche, forest fire, fountain, *etc.*, which carries more semantics compared with local image descriptors of interesting points or subregions. Moreover, we introduce a Laplacian regularizer in the objective function of sparse coding by modeling the semantic connections between the scenes in the dictionary to enforce not only sparsity but also semantic consistency in the generated video representations.

We conducted comprehensive experiments on two public dynamic scene recognition datasets, namely Maryland [2] and YUPENN [1]. The experimental results show that our proposed LSCoS works well with linear SVMs and outperforms the state-of-the-art techniques in video categorization. Moreover, it is worth mentioning that the proposed video representation can be applied to many other video applications as well, such as content-based similarity search.

The rest of the paper is organized as follows. The important related work is reported in Section II, followed by the technical details of our proposed method introduced in Section IV. Experimental results on model verification and comparison with the state-of-the-art methods are reported in Section IV. Finally, Section V concludes and suggests future work.

II. Related Work and Preliminaries

While the early studies on natural scene classification have majorly centered on still images [17], recent research has been seeking improved representation of scenes from dynamic videos [2], [1], [6]. Shroff *et al.* [2] characterized the dynamics of unconstrained scenes via chaotic systems. In the experiments, they presented the Maryland "in-the-wild" dataset, which contains 13 classes with 10 videos per class. By fusing the static and dynamic attributes, the classification rates were further improved. A larger dataset named YUPENN was introduced later by Derpanis *et al.* [1]. In their work, spacetime orientation measurements were investigated to describe image subregions. Similarly, Feichtenhofer *et al.* [6] presented a unified framework to extract bags of spacetime energies for dynamic scene recognition. The extracted features are next compressed by a novel dynamic max-pooling technique that has been shown to be more effective than the traditional max-pooling strategy. Inspired by the GIST descriptor [17] of images, Solmaz *et al.* [5] proposed a global video descriptor, GIST3D, which integrates the information about both the motion and the scene structure, for video classification. Although improved results have been reported, the majority of the literature only relies on the local features extracted from interest points or subregions. As pointed out by Penatti *et al.* [8], videos are composed by scenes that are usually with more semantic information than local features. The coding and pooling of scene-level descriptors have not been thoroughly evaluated yet.

Among the image representations, the Bag-of-Visual-Words (BoVW) [7] is the most widely used model to

generate a compact image descriptor while largely preserving the discriminative power of local features. Let $\boldsymbol{X} = [\vec{x_1}, \vec{x_2}, ..., \vec{x_m}]$ denote a set of local features of an image. BoVW generates a new code $\vec{z_i}$ for every feature $\vec{x_i} \in \boldsymbol{X}$ with a dictionary of basis vectors: $\boldsymbol{S} = [\vec{s_1}, \vec{s_2}, ..., \vec{s_n}]$. Let $z_{ij} \in \vec{z_i}$ be the coefficient with respect to $\vec{s_j} \in \boldsymbol{S}$, the commonly used coding schemes are summarized as follows. *Hard assignment* [10]: It quantizes feature $\vec{x_i}$ to its nearest neighbor in the dictionary. In most of the cases, the Euclidean distance is adopted,

$$z_{ij} = \begin{cases} 1 & \text{if} \quad j = \arg\min_j \|\vec{x_i} - \vec{s_j}\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

Soft assignment [11]: It computes z_{ij} based on the normalized distance between feature $\vec{x_i}$ and basis $\vec{s_j}$. The equation is given below where α is a smoothing factor that controls the softness of the assignment.

$$z_{ij} = \frac{\exp(-\alpha \|\vec{x_i} - \vec{s_j}\|_2^2)}{\sum_{k=1}^{n} \exp(-\alpha \|\vec{x_i} - \vec{s_k}\|_2^2)}$$

Sparse coding [12]: It approximates feature $\vec{x_i}$ with a linear sum of the basis vectors in the dictionary. The sparsity of $\vec{z_i}$ is enforced by the ℓ_1-regularization.

$$\vec{z_i} = \arg\min \|\vec{x_i} - \boldsymbol{S}\vec{z_i}\|_2^2 + \lambda \|\vec{z_i}\|_1$$

Inspired by the sparse coding, researchers have tried various regularization techniques to improve the coding effectiveness, such as *locality-constrained linear coding* [18], *low-rank sparse coding* [19], *Laplacian sparse coding* [13], *etc.* However, to the best of our knowledge, only the soft-assignment coding has been evaluated by Penatti *et al.* [8] for video geotagging. The effectiveness of the other coding techniques on scene-level descriptors still remains unknown for video processing.

III. Methodology

Here, we give a detailed description of our proposed framework to extract scene-level video representations.

A. Primitive Feature and Codebook Generation

As aforementioned, we process a video as a sequence of frames and extract global descriptors from every frame as the primitive features to be encoded. While previous methods capture the motion in videos by local features such as STIP [20] and BoSE [6], our approach describes the dynamics of scenes by coding the variations in the global visual appearances. This idea is motivated by the observation that global features usually carry certain semantics that local descriptors do not have, *e.g.*, the layout of a scene. Moreover, it becomes much more straightforward to associate the visual scenes in the dictionary with semantic labels, based on which the coding effectiveness can be significantly improved. The details of the coding scheme will be presented in Section III-B. Additionally in the experiments, we will see that the proposed method obtains excellent classification rates, even with low-level visual features such as color and HOG [16].

Next, we discuss the generation of the codebook, which is a set of basis vectors used as the dictionary for coding. Traditionally, k-means clustering is usually applied to group features based on the Euclidean distance in the visual space. Considering the basis vectors in our approach are visual scenes with semantic labels, the codebook can be further balanced according to the distribution of scenes in the semantic space. For video classification, frames are naturally associated with class names. To construct a dictionary of scenes, we first group the training data by their class tag, and then generate the same number (or proportional to the data distribution in the training set) of visual scenes from each class. In addition to k-means clustering, a simple random sampling scheme can be an alternative option due to its high efficiency and similar effectiveness [8]. A manual selection of representatives is also possible when the dictionary size is relatively small or can be reused by other applications like the ImageNet [21].

B. Laplacian Sparse Coding of Scenes

For a video consisting of m frames, primitive features (*e.g.*, color and HOG) are extracted and early fused into a single vector for every frame, denoted as $\boldsymbol{X} = [\vec{x_1}, \vec{x_2}, ..., \vec{x_m}]$ ($\vec{x_i} \in \mathbb{R}^{d \times 1}$) where d represents the feature dimension. Let $\boldsymbol{S} = [\vec{s_1}, \vec{s_2}, ..., \vec{s_n}] \in \mathbb{R}^{d \times n}$ be the codebook generated as described in the previous section. Based on the class labels associated with frames, we calculate a matrix W to describe the connections between the visual scenes in the codebook. Formally, we have,

$$w_{ij} = \begin{cases} 1 & \text{if } \vec{s_i} \text{ and } \vec{s_j} \text{ belong to the same class} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Intuitively, in terms of coding, the coefficients assigned to the visual scenes should be consistent with the semantic connections defined by W. That is to say, for a feature $\vec{x_i}$ to be encoded, if it shows a high degree of membership with respect to one visual scene in the dictionary, the coefficients assigned to other visual scenes with the same class label should also be promoted. However, as the visual appearance of a concept can vary a lot due to the semantic gap, the above semantic consistency cannot be guaranteed by traditional coding techniques that only focus on the visual space. Therefore, we propose an advanced video representation, Laplacian sparse coding of scenes, and formulate the objective function as follows,

$$\arg\min_{\boldsymbol{Z}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{S}\boldsymbol{Z}\|_F^2 + \lambda_1 \|\boldsymbol{Z}\|_{1,1} + \lambda_2 \, \mathit{trace}(\boldsymbol{Z}^\intercal L \boldsymbol{Z}) \tag{2}$$

where $\boldsymbol{Z} = [\vec{z_1}, \vec{z_2}, ..., \vec{z_m}]$ and $\vec{z_i} \in \mathbb{R}^{n\times 1}$ is the resulting representation for $\vec{x_i}$ after optimization. $\|\boldsymbol{Z}\|_{1,1} = \sum_{i=1}^{m} \|\vec{z_i}\|$ is the ℓ_1 regularization term to induce sparsity, which has been shown to be highly effective in feature quantization [12], [19]. *trace*$(\boldsymbol{Z}^\intercal L \boldsymbol{Z})$ is the Laplacian regularization term to encourage the pursuit representation matrix $\boldsymbol{Z}$ to be more consistent with the semantic connections among the visual scenes in the codebook, where L is the corresponding Laplacian matrix of W. By definition, L is computed as $L = D - W$ where D is a diagonal matrix with the i-th entry $d_{ii} = \sum_j w_{ij}$.

By solving the above optimization problem, the semantic consistency of sparse codes are significantly improved among the frames with similar content, leading to the generation of more descriptive video representations for downstream applications such as classification.

C. Optimization

We adopt the Inexact Augmented Lagrange Multiplier (IALM) [22] method to optimize the objective function. By adding one equality constraint, we convert Eq. 2 to the following equivalent format:

$$\begin{aligned} \arg\min_{\boldsymbol{Z_1}, \boldsymbol{Z_2}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{S}\boldsymbol{Z_1}\|_F^2 + \lambda_1 \|\boldsymbol{Z_2}\|_{1,1} + \lambda_2 \, trace(\boldsymbol{Z_1}^\intercal L \boldsymbol{Z_1}) \\ \text{s.t.} \boldsymbol{Z_1} = \boldsymbol{Z_2} \end{aligned} \tag{3}$$

where a slack variable $\boldsymbol{Z_2}$ is introduced so that we can handle the non-smooth regularizer $\|\boldsymbol{Z_2}\|_{1,1}$ separately. In order to solve Eq. 3, we introduce augmented Lagrange multipliers to incorporate the equality constraint into the objective function and obtain Eq. 4,

$$\begin{aligned} \arg\min_{\boldsymbol{Z_1}, \boldsymbol{Z_2}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{S}\boldsymbol{Z_1}\|_F^2 + \lambda_1 \|\boldsymbol{Z_2}\|_{1,1} + \lambda_2 \, trace(\boldsymbol{Z_1}^\intercal L \boldsymbol{Z_1}) \\ + trace(\boldsymbol{Y}^\intercal(\boldsymbol{Z_1} - \boldsymbol{Z_2})) + \frac{\mu}{2}\|\boldsymbol{Z_1} - \boldsymbol{Z_2}\|_F^2 \end{aligned} \tag{4}$$

where $\boldsymbol{Y}$ represents the Lagrange multipliers and $\mu > 0$ is a penalty parameter. Thereafter, we apply IALM, which is an iterative optimization algorithm, to solve Eq. 4 by updating $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$ one-at-a-time. Totally, the method consists of three update steps presented as below:

First, we update $\boldsymbol{Z_2}$ by solving the following equation derived based on the proximal gradient descent algorithm,

$$\begin{aligned} \boldsymbol{Z_2^*} &= \arg\min_{\boldsymbol{Z_2}} \lambda_1 \|\boldsymbol{Z_2}\|_{1,1} \\ &\quad + trace(\boldsymbol{Y}^\intercal(\boldsymbol{Z_1} - \boldsymbol{Z_2})) + \frac{\mu}{2}\|\boldsymbol{Z_1} - \boldsymbol{Z_2}\|_F^2 \\ &= \arg\min_{\boldsymbol{Z_2}} \frac{\lambda_1}{\mu}\|\boldsymbol{Z_2}\|_{1,1} + \frac{1}{2}\|\boldsymbol{Z_2} - (\boldsymbol{Z_1} + \frac{1}{\mu}\boldsymbol{Y})\|_F^2 \\ &= \mathcal{S}_{\frac{\lambda_1}{\mu}}(\boldsymbol{Z_1} + \frac{1}{\mu}\boldsymbol{Y}) \end{aligned} \tag{5}$$

where $\mathcal{S}_{\frac{\lambda_1}{\mu}}(\boldsymbol{A})$ is the element-wise soft-thresholding operator. Let $a_{ij} \in \boldsymbol{A}$ be an element in matrix $\boldsymbol{A}$, then $\mathcal{S}_{\frac{\lambda_1}{\mu}}(a_{ij}) = \text{sign}(a_{ij}) \max(0, |a_{ij}| - \frac{\lambda_1}{\mu})$.

Next, we update $\boldsymbol{Z_1}$ by solving Eq. 6, which is a smooth convex function that can be easily optimized as,

$$\begin{aligned} \boldsymbol{Z_1^*} &= \arg\min_{\boldsymbol{Z_1}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{S}\boldsymbol{Z_1}\|_F^2 + \lambda_2 \, trace(\boldsymbol{Z_1}^\intercal L \boldsymbol{Z_1}) \\ &\quad + trace(\boldsymbol{Y}^\intercal(\boldsymbol{Z_1} - \boldsymbol{Z_2})) + \frac{\mu}{2}\|\boldsymbol{Z_1} - \boldsymbol{Z_2}\|_F^2 \\ &= (\boldsymbol{S}^\intercal\boldsymbol{S} + \lambda_2\boldsymbol{L} + \lambda_2\boldsymbol{L}^\intercal + \mu\boldsymbol{I})^{-1}(\boldsymbol{S}^\intercal\boldsymbol{X} + \mu\boldsymbol{Z_2} - \boldsymbol{Y}) \end{aligned} \tag{6}$$

Finally, we update $\boldsymbol{Y}$ and μ by Eq. 7, where $\rho > 1$ is a constant that controls the increasing rate of μ and μ_{max} is a user-defined max threshold of μ.

$$\begin{cases} \boldsymbol{Y} = \boldsymbol{Y} + \mu(\boldsymbol{Z_1} - \boldsymbol{Z_2}) \\ \mu = \min(\rho\mu, \mu_{max}) \end{cases} \tag{7}$$

By repeating the above three steps to update the variables, we are able to optimize the objective function (Eq. 2) iteratively. The convergence of the algorithm is reached when the change in solution $\boldsymbol{Z}$ is below a user-defined threshold or a maximum number of iterations has been reached.

D. Max Pooling

The final stage is to generate a compact video representation by aggregating per-frame features while maximally preserving the descriptiveness. In order to achieve this goal, we adopt the max pooling strategy on the absolute sparse codes of $\boldsymbol{Z}$. Recall that $\boldsymbol{Z} = [\vec{z_1}, \vec{z_2}, ..., \vec{z_m}]$ and each column of $\boldsymbol{Z}$ represents the responses of a frame to the visual scenes in the dictionary. Let $\vec{u} = [u_1, u_2, ..., u_n]^\intercal$ be the final representation of the input video, then we have,

$$u_j = \max\{|z_{1j}|, |z_{2j}|, ..., |z_{mj}|\} \tag{8}$$

where $z_{ij} \in \vec{z_i}$ is the j-th element in vector $\vec{z_i}$. We choose max pooling due to its excellent performance in image and video categorization. It has been reported that max pooling outperforms other alternative pooling strategies and obtains the state-of-the-art classification effectiveness [12], [8].

E. Global and Local Scene Dynamics

As aforementioned in Section III-A, the input to our framework are global descriptors of images. We extract and encode features from frames, and capture the global dynamics of a video by the variations in the visual appearances of scenes. From careful examination we observe that the same feature extraction and sparse coding procedure can also be applied to the difference images between consecutive frames that capture the pixel-wise local dynamics. Therefore, in addition to the feature vector $\vec{u_g}$ calculated based on the original frame descriptors, we also compute $\vec{u_l}$ with the input being the difference images obtained by the subtraction of

consecutive frames. By combining $\vec{u_g}$ and $\vec{u_l}$, we arrive at the final video representation as,

$$\vec{\tilde{u}} = [w\vec{u_g}^\intercal, (1-w)\vec{u_l}^\intercal]^\intercal \quad (9)$$

where w is a balancing factor set to 0.5 by default. Note that the descriptiveness of $\vec{u_l}$ can be susceptible to the motion of the camera. With a fixed camera, consecutive frames subtraction effectively captures the foreground motion, leading to the representative $\vec{u_l}$ of the video. On the other hand, when a camera moves frequently with high speed, the use of $\vec{u_l}$ becomes less effective and therefore the balancing factor w should be set to a higher value.

IV. Experimental Results

We introduce the experimental setup in Section IV-A, and then proceed with the evaluations in two steps. First, we verify our model effectiveness through a comparison with other popular feature coding techniques. Next, we compare the proposed LSCoS with the state-of-the-art video representations in dynamic scene classification.

A. Experimental Setup

We have evaluated the proposed Laplacian Sparse Coding of Scenes (LSCoS) on two public dynamic scene recognition datasets Maryland [2] and YUPENN [1]. Videos with different illuminations, resolutions and camera dynamics were collected for a variety of scenes. A leave-one-video-out experiment was conducted to be consistent with previous work [4], [3], [6], [23]. The codebook of visual scenes was formed from online images. We collected 40 instances for each class, and hence the codebook size for datasets Maryland and YUPENN was 520 (13×40) and 560 (14×40), respectively. We set $\lambda_1 = 0.15$ and $\lambda_2 = 0.1$ in Eq. 2. The balancing factor w in Eq. 9 was set to 0.7 and 0.5 for Maryland and YUPENN, respectively, as the former dataset contains large camera motions.

In terms of primitive features, we extracted the following three image descriptors for our evaluations:

- **496-D HOG Descriptor:** Histograms of oriented gradients extracted over 4×4 fixed grid partitions [16].
- **496-D Color Descriptor:** Hue histograms extracted over 4×4 fixed grid partitions [14].
- **4096-D Deep Feature:** Output of the seventh (fc7) fully connected layer of a Convolutional Neural Network (CNN). Here we adopt the pre-trained Hybrid-CNN model [24], [23].

Frames were sampled at a rate of five per second. A multi-class linear SVM [12] was adopted for training and testing. All experiments were conducted using Octave on a server with two Intel® Xeon® E5-2680 v3 2.5GHz CPUs and 512 GB RAM. The runtime of our proposed method is less than 0.5 second per video in terms of feature encoding, with the threshold of the change in $\boldsymbol{Z}$ set to 10^{-3}.

B. Comparison with Popular Coding Techniques

We have compared our proposed method with the following three popular feature aggregation techniques and report the results:

- **Max:** Statistical measurement that applies max pooling directly on frame descriptors for temporal aggregation [23].
- **BoS:** Bag-of-Scene video representation that adopts soft-assignment as the coding scheme [8].
- **SC:** Compact video descriptor generated by sparse coding without the Laplacian regularizer in Eq. 2 [12].

To ensure a fair comparison, we used the same dictionary of visual scenes in all methods. The average classification accuracy on Maryland and YUPENN is shown in Tables I and II, respectively. For ease of comparison, we highlight the **best** and the second best results in the tables. As can be seen, the proposed LSCoS encoding outperforms its competitors BoS and SC in all cases. BoS adopts soft assignments where good performance can be obtained by using SVMs with nonlinear Mercer kernels, *e.g.*, the histogram intersection kernel [25]. However, for efficiency concerns when handling large datasets, linear SVMs are far more favored as they enjoy both much faster training and testing speeds [12]. To solve the above issue, SC was proposed in order to generate a sparse feature vector that works more effectively with simple linear SVMs. Compared with our method LSCoS, SC ignores the relationship among the basis vectors in the codebook, leading to a relatively worse classification accuracy. LSCoS, on the other hand, tries to ensure semantic consistency in video representations by introducing a Laplacian regularizer in the objective function, and therefore achieves state-of-the-art performance in video categorization problems.

In addition to the above advanced coding techniques, we also report the results obtained by a simple temporal aggregation of frame descriptors. The feature vector generated by Max has the same dimension as the input frame descriptor. As can be seen, Max with only low-level visual features of HOG and color is not able to achieve satisfactory classification results due to the semantic gap. With more robust image descriptors such as the CNN deep feature that has been trained on extensive image collections, considerable improvements are presented on both datasets. Compared with Max, our method LSCoS performs competitively well or better, with a performance gain as large as 8.5% in terms of classification accuracy. Moreover, by a simple late fusion of Max and LSCoS with equal weights, the best classification results have been obtained in all cases with different combinations of datasets and frame descriptors.

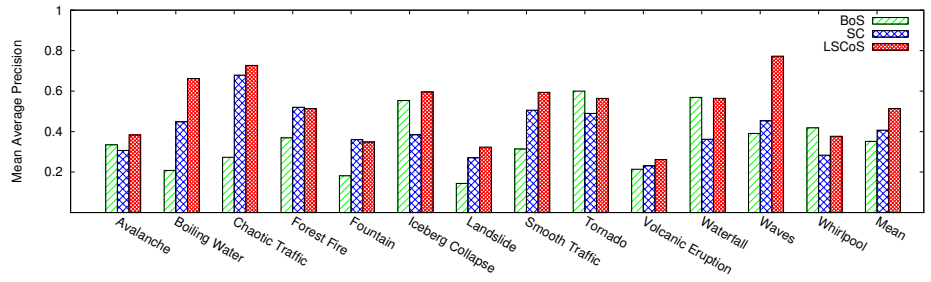
The main contribution of our work is to generate representations that preserve the semantic similarities among videos. To illustrate, we additionally performed a content-based similarity search experiment by using each image as a

Table I
AVERAGE CLASSIFICATION ACCURACY WITH DIFFERENT FEATURE AGGREGATION METHODS ON MARYLAND DATASET.

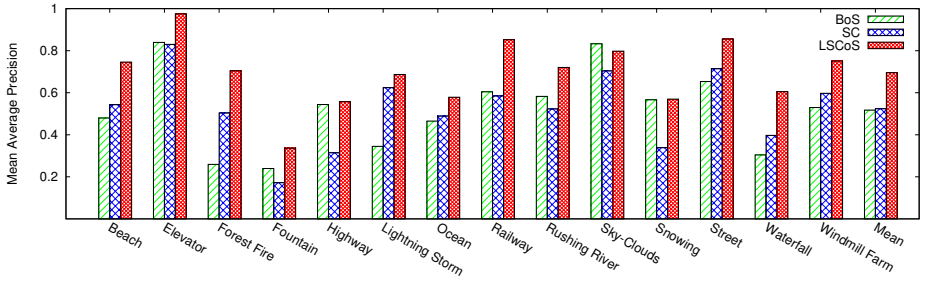
HOG+Color					Deep Feature				
Max	BoS	SC	LSCoS	Max+ LSCoS	Max	BoS	SC	LSCoS	Max+ LSCoS
62.31	57.70	64.62	70.77	**71.54**	93.85	80.77	90.77	93.08	**94.62**

Table II
AVERAGE CLASSIFICATION ACCURACY WITH DIFFERENT FEATURE AGGREGATION METHODS ON YUPENN DATASET.

HOG+Color					Deep Feature				
Max	BoS	SC	LSCoS	Max+ LSCoS	Max	BoS	SC	LSCoS	Max+ LSCoS
85.24	82.38	92.14	93.57	**94.29**	97.62	95.24	98.57	**99.05**	**99.05**



(a) Maryland dataset



(b) YUPENN dataset

Figure 3. Mean average precision comparison per class on the two public datasets using the deep feature.

Table III
MEAN AVERAGE PRECISION COMPARISON OF SIMILARITY SEARCH WITH DIFFERENT FEATURE AGGREGATION METHODS ON THE TWO PUBLIC DATASETS.

(a) Maryland dataset

	BoS	SC	LSCoS
HOG+Color	0.309	0.263	0.303
Deep Feature	0.352	0.407	0.514

(b) YUPENN dataset

	BoS	SC	LSCoS
HOG+Color	0.306	0.395	0.491
Deep Feature	0.518	0.524	0.695

query and rank the remaining images based on the Euclidean distance. Images from the same class as the query are considered to be relevant instances. We compare the ranking results and report the mean average precision obtained by different coding techniques in Table III. Moreover, a detailed per-class comparison is illustrated in Fig. 3. Our method LSCoS achieves the best mean average precision in most of the cases. It indicates that LSCoS generates similar representations for videos of the same class and effectively reduces the visual distance between relevant instances. Comparatively, BoS and SC do not preserve the semantic similarity of the original videos. The responses from similar video content using such techniques can vary significantly due to the semantic gap and the sensitiveness of feature quantization, resulting in less effective performance in content-based similarity search.

C. Comparison with the State-of-the-art

We compared our proposed technique with other state-of-the-art video representations: HOF+GIST [26],

Table IV
COMPARISON OF CLASSIFICATION ACCURACY WITH THE STATE-OF-THE-ART VIDEO REPRESENTATIONS ON MARYLAND DATASET.

Class	HOF+ GIST	Chaos+ GIST	SOE	SFA	CSO	GIST3D	BoSE	Max+ LSCoS (HC)	Max+ LSCoS (DF)
Avalanche	20	60	40	60	60	30	60	100	90
Boiling Water	50	60	50	70	80	60	70	80	90
Chaotic Traffic	30	70	60	80	90	70	90	50	100
Forest Fire	50	60	10	10	80	20	90	80	100
Fountain	20	60	50	50	80	20	70	90	100
Iceberg Collapse	20	50	40	60	60	50	60	60	100
Landslide	20	30	20	60	30	50	60	40	90
Smooth Traffic	30	50	30	50	50	40	70	80	90
Tornado	40	80	70	70	80	80	90	70	90
Volcanic Eruption	20	70	10	80	70	60	80	80	90
Waterfall	20	40	60	50	50	20	100	60	100
Waves	80	80	50	60	80	60	90	90	90
Whirlpool	30	50	70	80	70	50	80	50	100
Average	33.08	58.46	43.08	60.00	67.69	46.92	77.69	71.54	**94.62**

Table V
COMPARISON OF CLASSIFICATION ACCURACY WITH THE STATE-OF-THE-ART VIDEO REPRESENTATIONS ON YUPENN DATASET.

Class	HOF+ GIST	Chaos+ GIST	SOE	SFA	CSO	GIST3D	BoSE	Max+ LSCoS (HC)	Max+ LSCoS (DF)
Beach	87	30	93	93	100	90	100	97	97
Elevator	87	47	100	97	100	97	97	100	100
Forest Fire	63	17	67	70	83	83	93	93	100
Fountain	43	3	43	57	47	67	87	77	100
Highway	47	23	70	93	73	77	100	97	100
Lightning Storm	63	37	77	87	93	90	97	97	100
Ocean	97	43	100	100	90	100	100	100	100
Railway	83	7	80	93	93	97	100	93	100
Rushing River	77	10	93	87	97	93	97	100	100
Sky-Clouds	87	47	83	93	100	93	97	93	100
Snowing	47	10	87	70	57	77	97	90	93
Street	77	17	90	97	97	90	100	97	100
Waterfall	47	10	63	73	77	77	83	87	97
Windmill Farm	53	17	83	87	93	93	100	100	100
Average	68.33	22.86	80.71	85.48	85.95	87.38	96.19	94.29	**99.05**

Chaos+GIST [2], spatiotemporal oriented energies (SOE) [1], slow feature analysis (SFA) [3], complementary spacetime orientation (CSO) [4], GIST3D [5], and bags of spacetime energies (BoSE) [6].

The comparison of the classification effectiveness is reported in Tables IV and V. The last two columns are the results obtained by our proposed method with HOG+color (HC) and deep feature (DF), respectively. Even with low-level image descriptors, our method obtains promising classification rates and outperforms most of the previous techniques except BoSE. BoSE extracts and pools spacetime energy features from temporal instances (a sequence of frames). As a test video usually contains multiple temporal instances, the overall classification result is yielded by majority voting of all the temporal class predictions. Comparatively, our method generates a single feature vector per video as the final representation, which is a more efficient solution as the instances for training and testing have been greatly reduced. Moreover, with the robust CNN deep feature, our method performs significantly better than the other state-of-the-art techniques. Compared with the second best, BoSE, an improvement of 16.93% and 2.86% has been obtained for datasets Maryland and YUPENN, respectively.

To summarize, our proposed method achieves the best classification rates of 94.62% and 99.05% on both datasets. By encoding the semantic relationship among the visual scenes in the codebook, we are able to generate more descriptive video representations and subsequently improve the performance of downstream video applications.

V. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach named Laplacian sparse coding of scenes to generate high-level video representations that are more consistent and robust. In the objective function, we incorporate a Laplacian regularizer with the sparse coding technique to simultaneously promote or demote the coefficients assigned to visual scenes of the same class. Subsequently, videos that are similar in the semantic space are more likely to have consistent visual representations as well. We evaluated our method in video categorization and compared to the state-of-the-art techniques. Experimental results show that our method outperforms its competitors and achieves the best classification accuracy of 94.62% and 99.05% on two standard, publicly available

dynamic scene datasets, namely Maryland and YUPENN, respectively.

Future investigation will be conducted on the codebook construction and optimization. Regarding the calculation of the Laplacian matrix in the objective function, we only consider the intraclass connections at this point. In the future we would like to take the interclass distance into consideration as well. Moreover, fusion of visual features other than color and HOG should be evaluated for further improvements. It is also worth mentioning that though this high-level video representation we proposed has only been verified for video classification and similarity search in this work, it can be applied to many other video applications as well, *e.g.*, content-based video geotagging.

ACKNOWLEDGMENT

REFERENCES

[1] K. Derpanis, M. Lecce, K. Daniilidis, and R. Wildes, "Dynamic Scene Understanding: The Role of Orientation Features in Space and Time in Scene Classification," in *CVPR*, 2012, pp. 1306–1313.

[2] N. Shroff, P. Turaga, and R. Chellappa, "Moving Vistas: Exploiting Motion for Describing Scenes," in *CVPR*, 2010, pp. 1911–1918.

[3] C. Theriault, N. Thome, and M. Cord, "Dynamic Scene Classification: Learning Motion Descriptors with Slow Features Analysis," in *CVPR*, 2013, pp. 2603–2610.

[4] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spacetime Forests with Complementary Features for Dynamic Scene Recognition," in *BMVC*, 2013.

[5] B. Solmaz, S. M. Assari, and M. Shah, "Classifying Web Videos Using a Global Video Descriptor," *Mach. Vision Appl.*, vol. 24, no. 7, pp. 1473–1485, 2013.

[6] C. Feichtenhofer, A. Pinz, and R. Wildes, "Bags of Spacetime Energies for Dynamic Scene Recognition," in *CVPR*, 2014, pp. 2681–2688.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–2.

[8] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres, "A Visual Approach for Video Geocoding Using Bag-of-scenes," in *ICMR*, 2012, pp. 53:1–53:8.

[9] Y. Yin, R. Thapliya, and R. Zimmermann, "Encoded semantic tree for automatic user profiling applied to personalized video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *CVPR*, 2006, pp. 2169–2178.

[11] J. van Gemert, J.-M. Geusebroek, C. Veenman, and A. Smeulders, "Kernel Codebooks for Scene Categorization," in *ECCV*, 2008, vol. 5304, pp. 696–709.

[12] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear Spatial Pyramid Matching using Sparse Coding for Image Classification," in *CVPR*, 2009, pp. 1794–1801.

[13] S. Gao, I. Tsang, L.-T. Chia, and P. Zhao, "Local Features are not Lonely - Laplacian Sparse Coding for Image Classification," in *CVPR*, 2010, pp. 3555–3561.

[14] J. Van De Weijer and C. Schmid, "Coloring Local Feature Extraction," in *ECCV*, 2006, pp. 334–348.

[15] Y. Yin, Y. Yu, and R. Zimmermann, "On generating content-oriented geo features for sensor-rich outdoor video search," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1760–1772, 2015.

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, 2005, pp. 886–893.

[17] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained Linear Coding for Image Classification," in *CVPR*, 2010, pp. 3360–3367.

[19] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Low-Rank Sparse Coding for Image Classification," in *ICCV*, 2013, pp. 281–288.

[20] I. Laptev, "On Space-Time Interest Points," *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-scale Hierarchical Image Database," in *CVPR*, 2009, pp. 248–255.

[22] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-rank Matrices," *arXiv:1009.5055*, 2010.

[23] A. Gangopadhyay, S. M. Tripathi, I. Jindal, and S. Raman, "SA-CNN: Dynamic Scene Classification using Convolutional Neural Networks," *arXiv:1502.05243*, 2015.

[24] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.

[25] J. van Gemert, C. Veenman, A. Smeulders, and J.-M. Geusebroek, "Visual Word Ambiguity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, 2010.

[26] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," in *CVPR*, 2009, pp. 2929–2936.