Towards Scalable Summarization of Consumer Videos Via Sparse Dictionary Selection

Yang Cong, *Member, IEEE*, Junsong Yuan, *Member, IEEE*, and Jiebo Luo, *Fellow, IEEE*

***Abstract*—The rapid growth of consumer videos requires an effective and efficient content summarization method to provide a user-friendly way to manage and browse the huge amount of video data. Compared with most previous methods that focus on sports and news videos, the summarization of personal videos is more challenging because of its unconstrained content and the lack of any pre-imposed video structures. We formulate video summarization as a novel dictionary selection problem using sparsity consistency, where a dictionary of key frames is selected such that the original video can be best reconstructed from this representative dictionary. An efficient global optimization algorithm is introduced to solve the dictionary selection model with the convergence rates as $O(1/K^2)$ (where K is the iteration counter), in contrast to traditional sub-gradient descent methods of $O(1/\sqrt{K})$. Our method provides a scalable solution for both key frame extraction and video skim generation, because one can select an arbitrary number of key frames to represent the original videos. Experiments on a human labeled benchmark dataset and comparisons to the state-of-the-art methods demonstrate the advantages of our algorithm.**

***Index Terms*—Group sparse, key frame, Lasso, scene analysis, video analysis, video skim, video summarization.**

I. Introduction

RECENTLY, the availability of digital videos has been growing at an exponential rate, e.g., an estimated 20 h of videos are uploaded every minute to YouTube [1], and in 2007, users in the United States alone were producing over four billion hours of video footage each week. Thus, users are increasingly in need of assistance in accessing digital video, such as systems for search, interactive browsing, retrieval, semantic storage, or compressing video content. Video summarization techniques, also called video abstraction, are mainly designed to meet these requirements by developing a condensed version of a full-length video through the identification of the most important and pertinent semantic content within the video. They complement the automatic video retrieval methods (i.e., search), especially when content-based indexing and retrieval of video sequences has only seen limited success.

Although video summarization has been an extensively studied problem in the literature, many previous methods mainly focus on the summarization of structured videos, such as news, sports, or surveillance videos. However, the summarization of consumer videos is much more challenging. First of all, consumer videos contain much less constrained contents. For example, they can capture moving objects in a dynamic background, often with arbitrary camera motions. Moreover, there is no pre-defined structure in the consumer videos. Compared to shot-based key frame selection, e.g., in news and sports videos, it is much more ambiguous and subjective to select the key frames in an unstructured consumer video. In fact, it has been noted in [2] that it is difficult to evaluate the summarization results of consumer videos, because the ground truth often depends on the subject.

To handle the above challenges in summarizing consumer videos, we propose a novel and principled solution. In the following, we will first briefly explain our idea, followed by the discussion of the related work.

A. Overview

We formulate the problem of video summarization. Suppose the size of the entire video database is n frames, we represent the database as $B = \{b_1, b_2, \ldots, b_n\}$, where each 2-D frame can be denoted by a 1-D feature vector using bag of words (BOW), Gist [3], CENTRIST [4], [5], or stacking the 2-D pixels into a 1-D vector. Thus, $b_i \in \mathbb{R}^d$ denotes the ith image with dimension d. Our intention is to extract an optimal subset $B' \subset B, B' \in \mathbb{R}^{d\times k}, k \ll n$ which covers all the essential video contents with the size k as small as possible. Generally, depending on the specific application, the video summarization can be formulated as either key frame extraction or video skim generation.

Key Frames: They are representative images extracted from the underlying source video. Here, the key frame set B'_k is defined as

$$B'_k = \mathcal{F}_k(B) = \{b_{r_1}, b_{r_2}, \ldots, b_{r_k}\} \tag{1}$$

where $\mathcal{F}_k(\cdot)$ denotes the key frame extraction procedure, $B'_k \subset B$ is the selected key frame set with a size of r_k, and $r_i \in \{1, 2, \ldots, n\}$.

Video Skims: They are a collection of video segments extracted from the original video. Video skim itself is a video clip, but of a significantly shorter duration. We can define the problem as

$$B'_S = \mathcal{F}_s(B) = E_1 \bigcup E_2 \cdots \bigcup E_s,\ \forall E_i \bigcap E_j = \emptyset \tag{2}$$

Manuscript received February 07, 2011; revised August 02, 2011; accepted August 16, 2011. Date of publication September 01, 2011; date of current version January 18, 2012. This work was done when C. Yang was a research fellow at Nanyang Technological University and was supported in part by the Nanyang Assistant Professorship (SUG M58040015) to Dr. J. Yuan and NSFC (61105013). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Changsheng Xu.

Y. Cong is with the Department of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, and also with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China (e-mail: congyang81@gmail.com).

J. Yuan is with the Department of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jsyuan@ntu.edu.sg).

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: jiebo.luo@gmail.com).

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/TMM.2011.2166951

1520-9210/$26.00 © 2011 IEEE

where $\mathcal{F}_s(\cdot)$ denotes the skim generation procedure, $E_i \subset B$ is the ith excerpt to be included in the skim, $E_i = \{b_{e_1^i}, b_{e_2^i}, \ldots, b_{e_s^i}\}$, and $e_j^i \in \{1, 2, \ldots, n\}$.

In this study, we can define two explicit requirements for video summarization framework:

- **Sparsity**: Although the summarization should cover the video content, the size of the extracted data set should be as small as possible. Thus, we have $k \ll n$, which means B' should be "sparsely" selected from B.
- **Low reconstruction error**: The original data set B can be reconstructed with high accuracy using the selected data set "Dictionary" B', i.e., B' is the most representative subset of B.

Since these requirements are similar to the properties of sparsity consistency of group Lasso, we design a dictionary selection model to simulate the procedure of video summarization, i.e., we consider the whole video set as the original feature pool B, then we select an optimal subset B' as the "dictionary" from the pool B under two constraints, sparsity and lower reconstruction error.

The main contributions of this paper reside in three aspects:

1) We convert the video summarization problem into a dictionary selection problem, and also propose a novel dictionary selection model for video summarization using sparsity consistency, which is robust to tuning parameters.
2) We introduce a global optimization algorithm to solve our dictionary selection model with convergence rate of $O(1/K^2)$ (K is the iteration counter), which is more efficient than traditional sub-gradient descent methods of $O(1/\sqrt{K})$.
3) We design a scalable framework for both key frame extraction and video skim generation in a unified framework. In contrast to most existing methods, which require presetting the number of key frame, our framework allows users to choose different numbers of key frames without incurring additional computational cost.

The rest of this paper is organized as follows: Section II presents the formulation of the problem and our dictionary selection model, Section III describes the implementation of our video summarization based on the above sparse dictionary selection model. Various experiments and comparisons are presented in Section IV. Finally, Section V concludes the paper.

B. Prior Work

As video summarization is an extensively studied topic, a detailed review of all previous work is beyond the scope of this paper. Interested readers can check [6] and [7] for a more comprehensive summary. As motioned in [6], much previous work in video summarization has been focused on sports and news videos, such as the TRECVID dataset [8]–[12]. These types of videos have well-defined temporal structures and characteristics, which facilitate key frame extraction. For example, shot-based key frame selection in sports and news videos is much less ambiguous given the context. On the other hand, consumer videos, such as wedding or birthday party videos, do not exhibit the same level of temporal structure. Therefore, key frame selection from consumer videos is more subjective and can be affected by many factors such as the sentimental values and observer association.

Among the previous studies on consumer videos, [13] found that camera motion often appears quite limited, and sequences of zooming-in followed by a relatively stable camera motion is usually a robust indicator for video summarization. In [14], an intelligent key frame extraction method is developed for video printing. It is based on a few features, including accumulative color histogram, color layout difference, camera motion estimation, moving object tracking, face detection, and audio event detection. In [2], a semantically meaningful video summarization method is proposed for personal video clips, together with a new consumer video database with human labeled ground truth. Additionally, key frame extraction is also related to consumer video summarization. For instance, [15] proposed a user attention model and [16] used mosaic-based image representation for video summarization. Unsupervised clustering is applied in [17] for key frame extraction. In [18], it proposed a relative activity measurement method for video summarization. In [19], key frames are selected via scene categorization, and similar ideas are also applied in [20]–[26]. Besides summarizing consumer videos, [27] condensed long video sequences by Ribbon carving for visual surveillance. In [28], multi-view videos are summarized concurrently by representing the summarization problem as a hypergraph labeling task.

II. Problem Formulation and Model Definition

The key element for our video summarization can be considered as a dictionary selection problem, i.e., how to select an optimal subset from the entire video frame pool under certain constraints. In this section, we introduce a new dictionary model using group Lasso [29]. The group Lasso is a kind of sparse pursuit technology, which includes cardinality sparsity [30] and tensor rank sparsity [31] as well. We address the problem of selecting the dictionary given an initial candidate pool $B = [b_1, b_2, \ldots, b_n] \in \mathbb{R}^{d \times n}$, where each column vector $b_i \in \mathbb{R}^d$ denotes a video frame represented as a feature vector. Our goal is to find an optimal subset to form the dictionary $B' = [b_{i_1}, b_{i_2}, \ldots, b_{i_k}] \in \mathbb{R}^{d \times k}$ where $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n\}$, such that the original set B can be accurately reconstructed by B' and the size of B' is as small as possible. A simple idea is to pick candidates randomly or regularly to build the dictionary; the latter is also called evenly spaced key frames in previous works [2]. Apparently, this approach does not make full use of all features in B. Also it may miss important candidates or include the noisy ones, which will affect the accuracy of the reconstruction. To avoid these problems, we present a principled method to select the dictionary. We illustrate the idea of dictionary selection in Fig. 1. We can formulate the problem as

$$\min_X : \frac{1}{2}\|B - BX\|_F^2 + \lambda\|X\|_1 \tag{3}$$

where $X \in \mathbb{R}^{n \times n}$ is the pursuit coefficient matrix; the frobenius norm $\|X\|_F$ is defined as $\|X\|_F := (\sum_{i,j} X_{ij}^2)^{1/2}$ and the l_1 norm is defined as $\|X\|_1 = \sum_{i,j} |X_{ij}|$. However, one limitation of (3) is that it tends to generate a solution of X close to I, leaving the first term of (3) to be zero and very sparse as well. Thus, we need to enforce the consistency of the sparsity on the solution, i.e., the solution needs to contain some "0" rows such

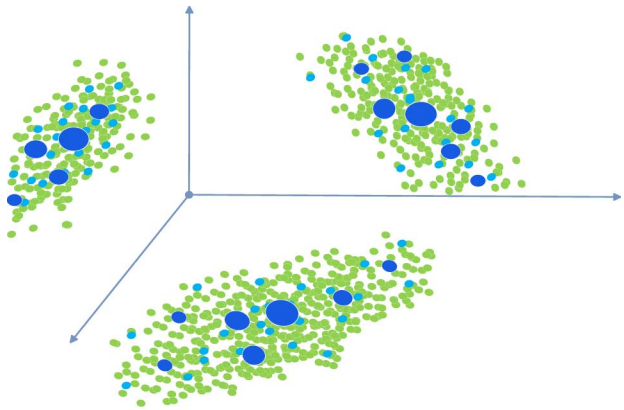

Fig. 1. Illustration of our algorithm. Imagine each video is sequentially connected by different scenes, each scene contains several frames and each frame is represented by a feature point in high dimensional space, so each scenario has its own distribution/cluster. For example, three scenes here are shown as three clusters, where the green points are feature points, the light blue points are sampled from each cluster and used to reconstructed individual frames in a scene. After dictionary selection, an optimal subset (dark blue points) is selected as the dictionary, where the size of each feature is related to the weight that measures the confidence of the selected feature being a key frame.

that the corresponding features in B are not selected to reconstruct any frame.

Therefore, it is better to change the l_1 norm constraint in (3) into the $l_{2,1}$ norm [32], and define the model as $\min_X : f(X) = 1/2\|B-BX\|_F^2+\lambda\|X\|_{2,1}$, where $\|X\|_{2,1} := \sum_i \|X_{i.}\|_2$, and $X_{i.}$ denotes the ith row of X. The $l_{2,1}$ norm is indeed a general version of the l_1 norm since if X is a vector, then $\|X\|_{2,1} = \|X\|_1$. In addition, $\|X\|_{2,1}$ is equivalent to $\|\mathbf{x}\|_1$ by constructing a new vector $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x}_i = \|X_{i.}\|_2$. The tuning parameter λ balances the weight of these two terms, i.e., construction error term and group sparsity term.

As it is difficult to select a suitable $\lambda \in [0\ \infty]$ in practical application, we further formulation our dictionary selection model as

$$\min_X : f(X) = \frac{\lambda}{2}\|B - BX\|_F^2 + \frac{(1-\lambda)}{2}\|X\|_{2,1} \qquad (4)$$

where $B \in \mathbb{R}^{d\times n}$ is the original feature pool; $X \in \mathbb{R}^{n\times n}$ is the pursuit coefficient matrix; and $\lambda \in [0\ 1]$ is the pre-set tuning parameters. The first term measures the quality, i.e., the reconstruction error, by using the selected dictionary to recover the whole feature pool. The second term of (4) denotes the sparse property of the dictionary selection. It follows that (4) leads to a sparse solution for X, i.e., X is sparse in terms of rows and the dictionary consists of features with $\|X_{i.}\|_2 \neq 0$. Based on (4), it is easy to balance the weights of these two terms by λ, where a smaller λ favors the second term (group sparsity term), while a larger λ prefers the first term (reconstruction error term), and $\lambda = 0$ or 1 are too extreme cases that should not happen in practice.

The solution of the optimization problem in (4) is a convex but nonsmooth optimization problem. Since $\|X\|_{2,1}$ is nonsmooth, although the general optimization algorithm (the subgradient descent algorithm) can solve it, the convergence rate is quite slow. Recently, Nesterov [33] proposes an algorithm to efficiently solve this type of convex (but nonsmooth) optimization problem and guarantees a convergence rate of $O(1/K^2)$ (K is the number of iterations), which is much faster than the general sub-gradient decent algorithm with a complexity of $O(1/\sqrt{K})$. We thus adopt Nesterov's method in [33] to solve (4). The approach proposed by Nesterov was also used to efficiently solve the trace norm minimization problem in machine learning [34].

Consider an objective function $f_0(x) + g(x)$ where $f_0(x)$ is convex and smooth and $g(x)$ is convex but nonsmooth. The key idea of Nesterov's method is to use $p_{Z,L}(x) := f_0(x) + \langle\nabla f_0(Z),x - Z\rangle + L/2\|x - Z\|_F^2 + g(Z)$ to approximate the original function $f(x)$ at the point Z. In each iteration, we need to solve $\arg\min_x : p_{Z,L}(x)$.

In our case, we have

$$\begin{cases} f_0(X) & = \frac{\lambda}{2}\|B - BX\|_F^2 \\ g(X) & = \frac{(1-\lambda)}{2}\|X\|_{2,1} \\ p_{Z,L}(X) & = f_0(Z) + \langle\nabla f_0(Z), X - Z\rangle \\ & \quad + \frac{L}{2}\|X - Z\|_F^2 + g(X). \end{cases} \qquad (5)$$

In order to solve (5), we can obtain the closed form solution as

$$\arg\min_X p_{Z,L}(X) = \mathcal{D}_{(1-\lambda)/2L}\left(Z - \frac{1}{L}\nabla f_0(Z)\right) \qquad (6)$$

where $D_\tau(.) : M \in \mathbb{R}^{n\times n} \mapsto N \in \mathbb{R}^{n\times n}$

$$N_{i.} = \begin{cases} 0, & \|M_{i.}\| \leq \tau; \\ \left(\frac{1-\tau}{\|M_{i.}\|}\right) M_{i.}, & \text{otherwise.} \end{cases} \qquad (7)$$

We will derive (6) in the Appendix, and the whole algorithm is presented in Algorithm 1.

Algorithm 1 Dictionary Selection

Input: B, $\lambda > 0$, K, X_0, c, L

Output: X

1: Initialize $Z_0 = X_0$, L, c, $a_0 = 1$.

2: **for** $k = 0, 1, 2, \ldots, K$ **do**

3: $X_{k+1} = \arg\min_X : p_{Z_k,L}(X) = D_{1-\lambda/2L}(Z - 1/L\nabla f_0(Z))$

4: **while** $f(X_{k+1}) > p_{Z_k,L}(X_{k+1})$ **do**

5: $L = L/c$

6: $X_{k+1} = \arg\min_X : p_{Z_k,L}(X) = D_{1-\lambda/2L}(Z - 1/L\nabla f_0(Z))$

7: **end while**

8: $a_{k+1} = (1 + \sqrt{1 + 4a_k^2})/2$

9: $Z_{k+1} = (a_{k+1} + a_k - 1/a_{k+1})\, X_{k+1} - (a_k - 1/a_{k+1})\, X_k$

10: **end for**

III. Implementation of Our Method

Given the above dictionary selection model, the framework of our video summarization algorithm is shown in Algorithm 2. We discuss the implementation details in this section.

Algorithm 2 General framework for Key frame selection

Input: the whole video set $B \in \mathbb{R}^{d\times n}$, λ

Output: the summarized set $B' \in \mathbb{R}^{d\times k}$

1: Pursuit the model $X^* = \arg\min_X \lambda/2\|B - BX\|_2^2 + (1-\lambda)/2\|X\|_{2,1}$

2: Generate weight curve using $\|X_{i.}\|_2$

3: Detect local maximums of weight curve and sort them

4: Extract key frames constrained by the weight value and full sequence coverage assumptions

A. Feature Representation

In our case, we intend to use a global feature to represent each frame, then collect all features to generate the candidate pool for dictionary selection. As consumer videos are usually composed of different scenes, we can adopt several features for scene understanding. Popular features are Gist [3] and CENTRIST [4], [5]. In this paper, we choose CENTRIST [Principal component Analysis of Census Transform (CT) histograms], which captures local structures of an image without color information. CENTRIST uses a spatial pyramid structure. In our case, we choose only the last two spatial levels, each including 5 and 1 image patches, respectively. Each patch is represented by a 42-dimensional feature, where 40 dimensions are for eigenvector and the other two are the mean and variance of the patch, respectively. Hence, the dimension of each CENTRIST feature is $6 \times 42 = 252$. More details can be found in [4] and [5].

Since CENTRIST does not contain color information, which is also very important for consumer video summarization, we choose HSV (hue, saturation, and brightness) color space and use color moment here as another descriptor. Color moment is based on the assumption that the distribution of color in an image can be interpreted as a probability distribution, and can be characterized by a number of unique moments. Stricker *et al.* [35] use three central moments of image's color distribution, i.e., mean, stand deviation, and skewness:

$$\begin{aligned} \text{Mean}: \quad & E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij} \\ \text{Standard Deviation}: \quad & \sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^2} \\ \text{Skewness}: \quad & s_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}(p_{ij}-E_i)^3} \end{aligned} \tag{8}$$

where p_{ij} denotes the ith color channel at the jth image pixel, N is the total number of pixels, and E_i, σ_i, and s_i are the mean, standard deviation, and skewness, respectively. Color moments are calculated from each color channel of an image patch; therefore, each image patch is characterized by 9 moments, i.e., 3 moments for each of the 3 color channels. In our implementation, we split an image into 3×4 patches, and each image patch is represented by 9 color moments, so each image is described by a color moment feature vector with a dimension of $3 \times 4 \times 9 = 108$.

We normalize the 252-D CENTRIST and the 108-D color moment, respectively, and stack them together to generate a combined feature vector with a dimension of $d = 252 + 108 = 360$.

B. Dictionary Selection

Given a video, we describe each frame as a global feature, and collect all frames to generate the feature pool, $B \in \mathbb{R}^{d\times n}$. Then we calculate $X \in \mathbb{R}^{n\times n}$ using our dictionary selection model in Algorithm 1, where each row of X corresponds to a feature, i.e., if the "weight" $\|X_{i.}\|_2$ is close to zero, the corresponding ith feature will not be selected; otherwise, the ith feature will be selected. This is the property of group sparsity and the selected features are used to build the dictionary for video summarization. We use three criteria to evaluate the model, 1) effectiveness: whether such a model can really sparsely select the dictionary; 2) robustness: whether we can obtain the same or similar dictionary using different parameter λ; 3) efficiency: whether the complexity is acceptable. All of these will be evaluated below.

Effectiveness: In (4), there is a tuning parameter λ used to balance the contributions of two different terms, reconstruction term and sparsity term. The smaller the value of λ, the more the sparsity term contributes to dictionary selection, which will make the dictionary sparser. As shown in Fig. 2, when given different parameter λ, we generate the "weight" curve, where the weight measures the confidence of the selected feature. We can see that the weights, $\|X_{i.}\|_2$, of most features are zeros, and the features with smaller weights are sparse, which validate the effectiveness of our model.

Robustness: It is important to check whether we can obtain similar or consistent results by selecting different λ. We illustrate it in Fig. 3. Each subfigure of Fig. 2 corresponds to each row of Fig. 3, where the brightness of each pixel corresponds to its own weight, i.e., the greater the value of weight is, the brighter the pixel is, and the value of each blue pixel is 0. As the value of λ decreases from bottom to up, the bright pixels in each row become much more sparse, because some unimportant features will be discarded from the dictionary. Note that there are some bright vertical lines, which means even if we tune the value of λ within a wide range, some important features can always be selected as a component of the dictionary. In other words, we can obtain stable dictionaries robustly even using different λ.

Efficiency: As mentioned above, the convergence rate of our algorithm is only $O(1/K^2)$ (K is the number of iterations), which is much more efficient than that of the general sub-gradient descent algorithms ($O(1/\sqrt{K})$). Given $B \in \mathbb{R}^{360\times 600}$ and by tuning λ, we can obtain the convergence times versus reconstruction cost curve as shown in Fig. 4. It can be found that when $\lambda = 0.33$, it converges after less than 500 iterations,

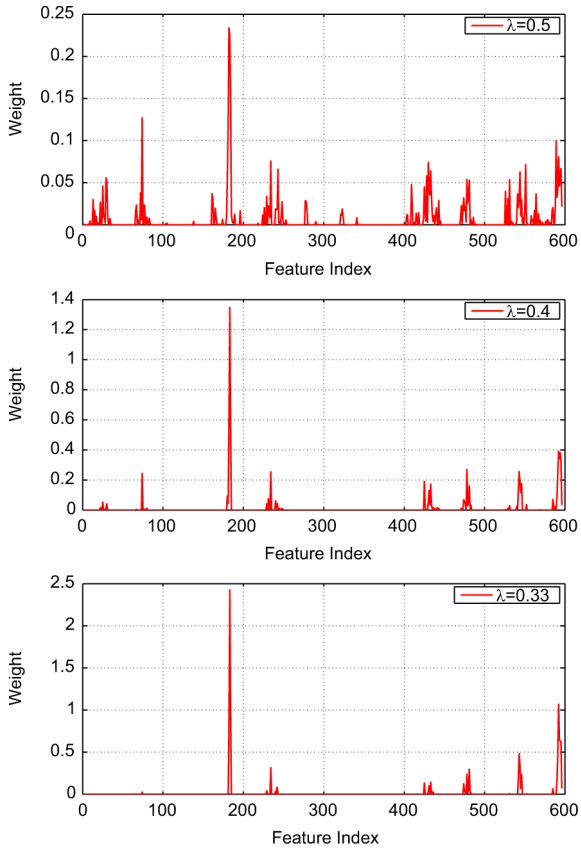

Fig. 2. Effectiveness of the algorithm. The horizontal axis is the feature index, the vertical axis is the weights corresponding to each feature as $\|X_i.\|_2$, where i is the row index. The smaller the value of λ is, the more sparse the result is. Note that the features with low or zero weights cannot be inserted into the dictionary.

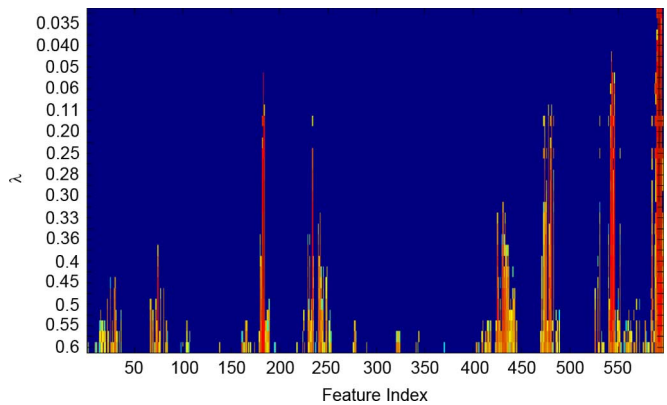

Fig. 3. Robustness of the algorithm. The horizontal axis is the feature index, and the vertical axis is the value of λ, so each row can be plotted as a subfigure in Fig. 2. A brighter value means a larger weight, and the blue value means 0, i.e., if a feature is selected as the feature of dictionary, the value should be bright, and the blue means discarded otherwise. Two points can be concluded, 1) the smaller the λ value, the more sparse the results is; 2) for different values, our algorithm can select the same stable features as the dictionary, which is essential to validate the robustness of our algorithm.

and when $\lambda = 0.5$, it converges after about 1000 iterations. In contrast, the traditional sub-gradient descend algorithms with

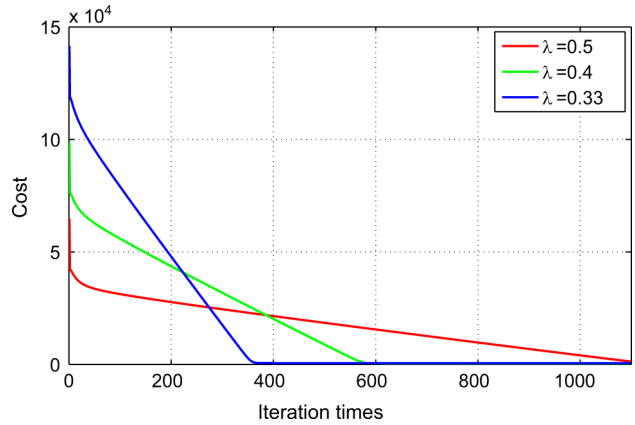

Fig. 4. Efficiency of the algorithm. The horizontal axis is the iteration times and the vertical axis is the $cost = \lambda\|X\|_{2,1}$ for group Lasso. From $\lambda = 0.33$ to 0.5, the smaller the value of λ, the faster the convergence speed. Regardless, even when $\lambda = 0.5$, it only takes 1000 iterations to converge, which is much faster than traditional gradient descent optimization algorithm requiring nearly 2 000 000 iterations with the same λ.

the same configuration ($\lambda = 0.5$) will converge after about 2 000 000 iterations.

C. Video Summarization

After dictionary selection from the candidate pool, we obtain a "weight" curve. Each weight corresponds to the L_2 norm of each feature. Then the key frame and video skim can be extracted based on these weight, as shown in Figs. 5 and 6. Comparing to other video summarization algorithms, our algorithm mainly has two merits:

- Our algorithm is scalable, which provides flexibility for practical applications, i.e., given the "weight" curve of dictionary selection, users can change the parameters arbitrarily (such as the number of key frames) for selecting key frames or video skims without increasing additional complexity cost. In contrast, many other key frame extraction algorithms need to preset key frame number and any change will result in a re-calculation.
- Our algorithm provides a unified solution for both key frames extraction and video skims generation, as demonstrated in Fig. 5. For key frames extraction, we first detect all the local maxima and sort them according to weights. Then given the preset key frame number, key frames are extracted based on two constraints, i.e., the representativeness and coverage. For video skim generation, first, the length of each skim in a shot is calculated according to the number of key frames, skimming ratio, and shot duration L. The total skim length of a shot should be evenly distributed among all key frames. Also, any segment should not be shorter than the minimum length $L_{\min}$. For experiments in this paper, we only focus on key frame extraction.

An example of our key frame extraction procedure is shown in Fig. 6. Note, the key frames generated by our algorithm sometimes are somewhat different from but close enough to human labeled ground truth. According to our observation, the reason is that for each scene, humans tend to select the first frame from such a scenario as the key frame.

TABLE I
LABELED SET OF VIDEO CLIPS USED FOR EVALUATION

Video ♯	Video Name	♯KF	FPS	♯Frames	Indoor/Outdoor	Camera Motion	Object Motion	Perspective Changes	Bright. Changes
1	SchoolBand	6	30	1453	Indoor	Yes	Moving landscape	No	No
2	AsianExhibit	6	24	836	Indoor	Yes	Moving landscape	Yes	Yes
3	CardsAroundTable	7	24	343	Indoor	Yes	Moving ROIs	Yes	Yes
4	ReporterTour	9	24	1443	Indoor	Yes	Moving landscape & ROIs	Yes	Yes
5	SoloSurfer	6	24	618	Outdoor	Yes	Moving landscape	Yes	Yes
6	SkylineFromOverlook	6	24	559	Outdoor	Yes	Moving landscape	Yes	Yes
7	FireworksAndBoat	4	24	656	Outdoor	No/Yes	Moving ROIs	No	No
8	EatingSardines	6	24	437	Outdoor	Yes	Moving ROIs	Yes	No
9	BusTour	5	24	541	Outdoor	Yes	Moving landscape	Yes	Yes
10	LiquidChocolate	6	24	397	Indoor	Yes	No	Yes	Yes
11	OrnateChurch	4	24	194	Outdoor	Yes	Moving landscape	Yes	Yes
12	LawnchairDance	6	24	448	Outdoor	No/Yes	Moving ROIs	No	No
13	KnockKnockPrincess	7	24	2800	Indoor	No	Moving ROIs	No	No
14	Wedding	4	24	1380	Indoor	No	Moving ROIs	No	No
15	HappyDog	4	24	376	Outdoor	Yes	Moving ROIs	Yes	Yes
16	MuseumExhibit	4	24	250	Indoor	Yes	No	No	No
17	PoolAcrobat	7	24	1025	Outdoor	Yes	Moving landscape & ROIs	Yes	No
18	BoatUpsideDown	12	24	605	Outdoor	Yes	Moving landscape & ROIs	Yes	Yes

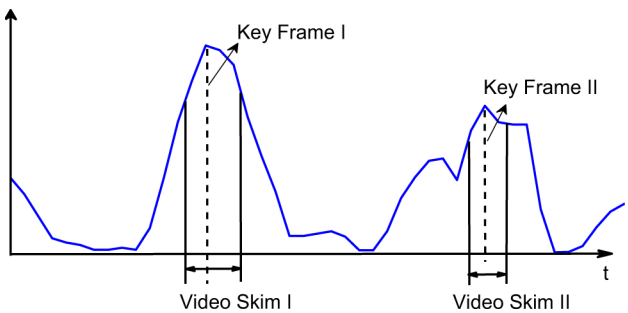

Fig. 5. Video summarization scheme, where the horizontal axis denotes the index of frame, the vertical axis is the "weight" of each feature. Depending on such curve, we can extract key frames or generate video skims.

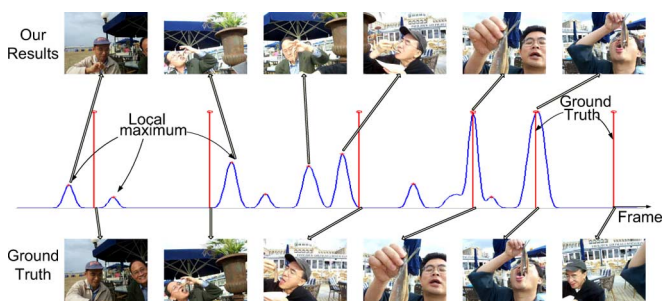

Fig. 6. Candidate extraction from a series of image sequences based on the weight. The top and bottom rows of images are generated by our algorithm and ground truth, respectively. We first detect local maxima from the weight curve, then extract key frames from local maximums constrained by the weight value and full sequence coverage assumption.

IV. EXPERIMENTS

In this section, we present various experiments and comparisons to validate the effectiveness and efficiency of our proposed algorithm.

Database: We choose the Kodak Home Video Database [2] for testing, instead of other video summarization databases that only contain sports or news video clips. There is subset of the consumer video clips in the Kodak Home Video Database with sufficient context and the ground truth labeled by several persons. The labeled database consists of 100 QuickTime clips selected from [36], which are captured using Kodak EasyShare C360 and V550 zoom digital cameras, with a VGA resolution and frame rates of 24 to 30 frames per second (FPS). For each clip, three judges selected key frames in order to build the ground truth. The number of key frames is not fixed for the judges so they can focus on representativeness and quality. A subset of 18 clips, including eight indoor and ten outdoor videos with a variety of scene content and camera motion, are selected for algorithm evaluation. Descriptions of these 18 clips are given in Table I. The average clip length is about 34 s and the average number of key frames is about six in the ground truth. For "wedding" sequence, we only have 1380 frames with 4 ground truth key frames, so the total number of key frame ground truth in our case is 109 instead of 113 in [2].

Evaluation: In order to quantitatively determine whether the predicted key frames and the corresponding ground truth are similar to each other, both image content and time differences are considered as suggested in [6], i.e., two frames must occur within a short period of time, and must be similar in scene content and composition to be considered equivalent. We also follow the assignment, i.e., the degree of a match is assigned 0, 0.5 and 1, where 0.5 corresponds to a weak match.

Comparison: We compare our proposed dictionary selection based video summarization (DSVS) algorithm [2] with several other methods, such as evenly spaced key frames (ESKF), color histogram-based method of UCF [19], motion-based key frame extraction method (MKFE) [2], and also online clustering key frames extraction (OCFE) using the same features as ours. The results are shown in Table II.

The **ESKF** generates the same number of key frames as ground truth by uniformly sampling the video along the time axis. The result is 45% (or 51.5/113 key frames), which is surprisingly good to match the ground truth approximately. Although ESKF may be considered fairly effective on average, the results do not consider the image content. The limitations are: 1) Due to lack of enough prior knowledge about a clip, it is impossible for us to predefine a correct number of key frames for summarization, and results will be quite different depending on the desired number of key frames; 2) the effectiveness of

TABLE II

SCORES (MATCHED KEY FRAMES WITH GROUND TRUTH) OBTAINED FOR THE 18 KODAK VIDEO DATASET. COLUMNS 1 AND 2 ARE THE INDEX AND VIDEO NAME, RESPECTIVELY. COLUMN 3: SCORES (IN RATIO AND PERCENTAGE) FOR THE SAME NUMBER OF EVENLY SPACED KEY FRAMES (ESKF). COLUMN 4: SCORES FOR THE MOTION-BASED KEY FRAMES EXTRACTION (MKFE) ALGORITHM [2]. COLUMN 5: SCORES FOR A COLOR HISTOGRAM-BASED ALGORITHM [19]. COLUMN 6: SCORES FOR THE ONLINE CLUSTERING BASED ALGORITHM (OCFE). COLUMN 7: SCORES FOR OUR PROPOSED DICTIONARY SELECTION-BASED KEY FRAMES EXTRACTION ALGORITHM (DSVS). COLUMNS 8, 9, AND 10 ARE FOR MKFE, OCFE, AND OUR DSVS WITH ONLY THREE KEY FRAMES. THE BOTTOM ROW ARE THE AVERAGE ACCURACY. OBVIOUSLY, OUR DSVS OUTPERFORMS OTHER METHODS

Video ♯	Video Name	Evenly spaced (ESKF) Score for a same ♯ of KF	Motion-based (MKFE) Score for a same ♯ of KF	Histogram-based (UCF) Score for a same ♯ of KF	Online Clustering (OCFE) Score for a same ♯ of KF	Dictionary Selection (DSVS) Score for a same ♯ of KF	Motion-based (MKFE) Nb of matched KF if only 3 Auto KF	Online Clustering (OCFE) Nb of matched KF if only 3 Auto KF	Dictionary Selection (DSVS) Nb of matched KF if only 3 Auto KF
1	SchoolBand	4/6(67%)	3/6(50%)	3/6(50%)	4.0/6(67%)	4.0/6(67%)	2/3(67%)	0.5/3(17%)	2.0/3(67%)
2	AsianExhibit	2/6(33%)	3/6(50%)	2/6(33%)	3.5/6(58%)	5.0/6(83%)	2/3(67%)	1.0/3(33%)	2.0/3(67%)
3	CardsAroundTable	4/7(57%)	4.5/7(64%)	5.5/7(78.5%)	4.0/7(57%)	6.0/7(86%)	2/3(67%)	2.0/3(67%)	3.0/3(100%)
4	ReporterTour	4/9(44%)	4.5/9(50%)	4/9(44%)	5.0/9(56%)	5.0/9(56%)	1/3(33%)	1.5/3(50%)	1.0/3(33%)
5	SoloSurfer	4/6(67%)	4/6(67%)	2/6(33%)	3.5/6(58%)	4.5/6(75%)	3/3(100%)	2.0/3(67%)	2.0/3(67%)
6	SkylineFromOverlook	5/6(83%)	3.5/6(58%)	4/6(66%)	4.5/6(75%)	5.0/6(83%)	1.5/3(50%)	3.0/3(100%)	2.0/3(67%)
7	FireworksAndBoat	0/4(0%)	3/4(75%)	0/4(0%)	2.0/4(50%)	3.0/4(75%)	2/3(67%)	1.5/3(50%)	3.0/3(100%)
8	EatingSardines	1.5/6(25%)	5/6(83%)	3/6(50%)	4.0/6(67%)	5/6(83%)	3/3(100%)	2.0/3(67%)	3.0/3(100%)
9	BusTour	2.5/5(50%)	2/5(40%)	3/5(60%)	2.0/5(40%)	3.0/5(60%)	2/3(67%)	1.0/3(33%)	2.0/3(67%)
10	LiquidChocolate	2/6(33%)	4/6(67%)	3/6(50%)	3.5/6(58%)	4.5/6(75%)	3/3(100%)	2.0/3(67%)	2.0/3(67%)
11	OrnateChurch	2/4(50%)	3/4(75%)	3/4(75%)	2.5/4(63%)	3.0/4(75%)	2/3(67%)	1.5/3(50%)	2.0/3(67%)
12	LawnchairDance	3/6(50%)	3/6(50%)	3/6(50%)	4.0/6(67%)	3.0/6(50%)	0/3(0%)	1.0/3(33%)	2.0/3(67%)
13	KnockKnockPrincess	1/7(14%)	2/7(28.5%)	2/7(28.5%)	2.5/7(36%)	2.5/7(36%)	0/3(0%)	0.0/3(0%)	0.5/3(17%)
14	Wedding	1/8(15.5%)	6/8(75%)	2/8(25%)	2.0/4(50%)	3.0/4(75%)	3/3(100%)	0.0/3(0%)	2.0/3(67%)
15	HappyDog	3/4(75%)	3/4(75%)	2/4(50%)	3.0/4(75%)	2.5/4(63%)	2/3(67%)	3.0/3(100%)	2.0/3(67%)
16	MuseumExhibit	2/4(50%)	3/4(75%)	2/4(50%)	2.0/4(50%)	2.0/4(50%)	3/3(100%)	1.5/3(50%)	1.5/3(50%)
17	PoolAcrobat	4/7(57%)	3/7(43%)	3/7(43%)	2.5/7(36%)	2.5/7(36%)	2/3(67%)	1.0/3(33%)	1.0/3(33%)
18	BoatUpsideDown	6.5/12(54%)	6.5/12(54%)	5.5/12(46%)	7.0/12(58%)	7.0/12(58%)	2.5/3(83%)	3.0/3(100%)	2.0/3(67%)
Total Average:		51.5/113 (45.5%)	64/113 (58.4%)	52.5/113 (46.5%)	61.5/109 (56.4%)	**70.5/109** (**64.7**%)	36/54 (67%)	27.5/54 (50.9%)	**35/54** (**64.8**%)

such a method is highly dependent on the video content, where ESKF would produce better matches for a pure pan segment than for a video with complex, nonlinear object motion. For example, clips number 7 (FireworksAndBoat), 12 (LawnchairDance), 13 (Wedding), and 14 (KnockKnockPrincess) exhibit mostly steady or fixed camera motion. Applying ESKF to these four sequences leads to an accuracy score of merely 5/25 (20%), while motion-based algorithm MKFE and our algorithm DSVS obtain better results.

The motion-based method (**MKFE**) is required to produce the same number of key frames as in the ground truth. The result is shown in Table II, which obtains about 64/113 (58.4%) accuracy.

The **UCF** algorithm [19] extracts key frames depending on the color histogram intersection similarity measure. In particular, the first frame is chosen by algorithm as the first key frame, and then additional ones are selected based on how it differs from existing ones. The threshold can decide how many key frames to be extracted, or the algorithm can automatically determine the number of frames that need to be extracted based on a similarity threshold. As shown in the fifth column of Table II, the UCF obtains an average accuracy of 52.5/112 (45.5%).

The **OCFE** is also chosen here, because our dictionary selection algorithm can also be considered as a clustering method. By using the same feature vector to represent each frame, the OCFE obtains an average accuracy of 61.5/109 (56.4%) which is lower than our proposed DSVS algorithm with an accuracy of 70.5/109 (64.7%), but is comparable to the motion-based method, MKFE, whose accuracy is 64/113 (58.4%). The reason is that our dictionary selection-based method using group sparsity consistency is highly discriminative in the high dimensional feature space.

Finally, the statistic result of our proposed **DSVS** algorithm is shown in the 7th column of Table II, which outperforms all other methods with an average accuracy of 70.5/109 (64.7%). Specially, we obtain near perfect results with large gaps to other methods on some video clips, such as "AsianExhibit", "CardsAroundTable", and "EatingSardines". Moreover, in comparison with MKFE, our method (DSVS) only adapts to the semantic content without being explicitly assisted by motion cues, which makes our DSVS faster for consumer video clips. Since our DSVS outperforms OCFE using the same type of feature, it confirms the effectiveness of our dictionary selection model. On the other hand, as a scalable method, our DSVS does not incur additional complexity cost when changing configurations.

The ability to produce any desired number of key frames and to supply labeled and ranked output frames is a desired feature for practical applications. In the last three columns of Table II, we demonstrate the effectiveness of our candidate key frame ranking procedure. It reports the number of matches when extracting only three key frames (the top three candidates having the highest confidence values). 35/52 (64.7%) of these frames correspond to true key frames in the ground truth.

Figs. 7–10 provide some examples of comparison between the ground truth, the motion-based method MKFE [2], the color histogram-based method courtesy of UCF [19], ESKF, OCEF, and our dictionary selection-based method DSVS, which illustrate the advantage of the proposed DSVS method over the other methods. Pictures with thick borders are considered good matches with ground truth, while dashed borders indicate weak matches ($score = 0.5$ instead of 1). In Fig. 7, while there are several different scenes, including stationary scene and moving children, our DSVS obtains the best result by a large margin over other methods. In Fig. 8, the camera is

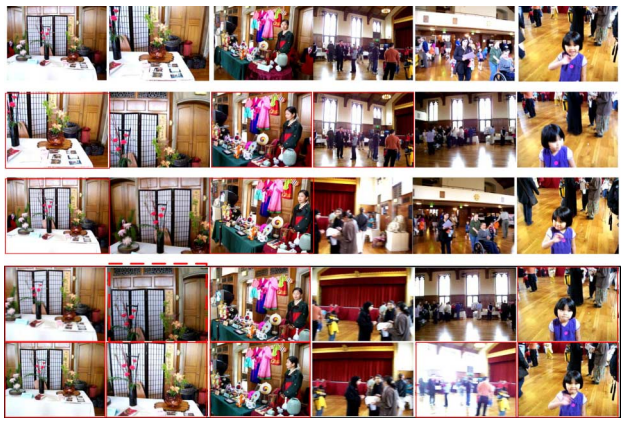

Fig. 7. Example "AsianExhibit". Comparison is between ground truth (first row), MKFE (second row), ESKF (third row), OCFE (fourth row), and our proposed DSVS (fifth row). Scores are 33% for ESKF, 50% for MKFE, 58% for OCFE, and 83% for our DSVS. Thick borders indicate good matches and dashed borders indicate subpar matches ($score = 0.5$ instead of 1).

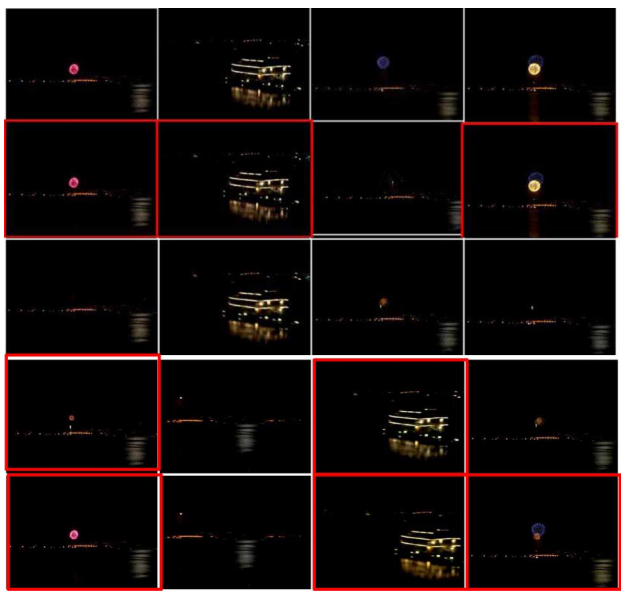

Fig. 8. Example "FireworkAndBoat". Comparison is between ground truth (first row), MKFE (second row), ESKF (third row), OCFE (forth row), and our proposed DSVS (fifth row). Scores are 0% for ESKF (second frame at bottom depicts similar scene but is too blurry), 75% for MKFE, 50% for OCEF, versus 75% for DSVS. Thick borders indicate good matches.

mostly steady, except for a rapid camera motion toward a boat. The key frames produced by MKFE are all representative of the high level object motion, and our DSVS generates similar results. In Fig. 9, the video was captured during an unsteady pan behind the window on a moving bus. The motion-based method partially failed because the algorithm could not compute the real displacement among violent perspective changes. Consequently, it did not succeed in providing successive key frames while minimizing spatial overlap in between. The histogram-based method, UCF, extracted key frames containing different content, which happens to work well for the case when the camera is mostly panning. Comparatively, our DSVS gets the same accuracy as UCF for this case. Fig. 10 is also

Fig. 9. "BusTour". Comparison is between ground truth (first row), MKFE (second row), UCF (third row), ESKF (fourth row), OCFE (fifth row), and our proposed DSVS (sixth row). Scores are 50% for ESKF, 60% for UCF, 40% for MKFE, 40% for OCFE, and 60% for DSVS. Thick borders indicate good matches and dashed borders indicate subpar matches ($score = 0.5$ instead of 1).

Fig. 10. "LiquidChocolate". Comparison is between ground truth (first row), MKFE (second row), UCF (third row), ESKF (fourth row), OCFE (fifth row), and our proposed DSVS (sixth row). Compare to our DSVS with score as 75%, scores are 33% for ESKF, 50% for UCF, 66% MKFE, and 58% for OCEF. Thick borders indicate good matches and dashed borders indicate subpar matches ($score = 0.5$ instead of 1).

quite interesting because this sequence contains three parts. The camera first fixes on an object (chocolate fountain) for nearly half of the sequence. Then, the camera scans the shop in a rapid pan before successive zooms (in and out) at the end. Instead of three frames generated by UCF and two frames produced by MKFE and ESKF, there is only one frame retained by the OCEF and our DSVS that summarizes the first part of the sequence. Moreover, two key frames associated with the successive zoom operations are correctly detected at the end of the sequence.

V. Conclusion

Although video summarization has been extensively studied for highly structured videos, such as sports and news videos, it is a much more challenging task to summarize consumer videos that are much less constrained and lack any pre-imposed structures. By representing each video frame as a feature vector, we convert the video summarization problem to a sparse dictionary selection problem. Our goal is to find a subset of key frames, i.e., the dictionary, such that it can best reconstruct the original video. A novel dictionary selection model (DSM) is proposed, together with an efficient global optimization algorithm, to iteratively refine the dictionary selection model with a convergence rate outperforming the traditional sub-gradient descent methods. As a video sequence can be summarized with any given number of key frames, our method provides a scalable solution to video summarization. We test our method on a human labeled benchmark dataset and compare it with the state-of-the-art methods. The results validate the advantages of our method.

Appendix

Proof: First the optimization problem $\min_X : p_{Z,L}(X)$ can be equivalently written as

$$\begin{aligned}
&\min_X : \ f_0(Z) + \langle \nabla f_0(Z), X - Z\rangle + \frac{L}{2}\|X - Z\|_F^2 + g(X) \\
&\Leftrightarrow \min_X : \ \frac{L}{2}\|(X - Z) + \frac{1}{L}\nabla f_0(Z))\|_F^2 + \frac{(1-\lambda)}{2}\|X\|_{2,1} \\
&\Leftrightarrow \min_X : \ \frac{L}{2}\|X - (Z - \frac{1}{L}\nabla f_0(Z))\|_F^2 + \frac{(1-\lambda)}{2}\|X\|_{2,1} \\
&\Leftrightarrow \min_X : \ \frac{L}{2}\|X - (Z - \frac{1}{L}\nabla f_0(Z))\|_F^2 + \frac{(1-\lambda)}{2}\sum_{i=1}^{k}\|X_{i.}\|_2.
\end{aligned} \tag{9}$$

Since the l_2 norm is self-dual, the problem above can be rewritten as by introducing a dual variable $Y \in \mathbb{R}^{k\times k}$:

$$\begin{aligned}
&\min_X : \frac{L}{2}\left\|X - \left(Z - \frac{1}{L}\nabla f_0(Z)\right)\right\|_F^2 \\
&\qquad + \frac{(1-\lambda)}{2}\sum_{i=1}^{k}\max_{\|Y_{i.}\|_2\le 1}\langle Y_{i.}, X_{i.}\rangle \\
&\Leftrightarrow \max_{\|Y_{i.}\|_2\le 1}\min_X : \frac{L}{2}\left\|X - \left(Z - \frac{1}{L}\nabla f_0(Z)\right)\right\|_F^2 \\
&\qquad + \frac{(1-\lambda)}{2}\sum_{i=1}^{k}\langle Y, X\rangle \\
&\Leftrightarrow \max_{\|Y_{i.}\|_2\le 1}\min_X : \frac{1}{2}\left\|X - \left(Z - \frac{1}{L}\nabla f_0(Z) - \frac{(1-\lambda)}{2L}Y\right)\right\|_F^2 \\
&\qquad - \frac{1}{2}\left\|Z - \frac{1}{L}\nabla f_0(Z) - \frac{(1-\lambda)}{2L}Y\right\|_F^2.
\end{aligned} \tag{10}$$

The second equation is obtained by swapping "max" and "min". Since the function is convex with respect to X and concave with respect to Y, this swapping does not change the problem according to the Von Neumann minimax theorem. Letting $X = Z - 1/L\nabla f_0(Z) - (1-\lambda)/2LY$, we obtain an equivalent problem from the last equation above:

$$\max_{\|Y_{i.}\|_2\le 1} : -\frac{1}{2}\left\|Z - \frac{1}{L}\nabla f_0(Z) - \frac{(1-\lambda)}{2L}Y\right\|_F^2. \tag{11}$$

Using the same substitution as above $Y = -2L/(1-\lambda)(X - Z + 1/L\nabla f_0(Z))$, we change it into a problem in terms of the original variable X as

$$\begin{aligned}
&\min_{\|2L/(1-\lambda)(X-Z+1/L\nabla f_0(Z))_{i.}\|_2\le 1} : \|X\|_F^2 \\
&\Leftrightarrow \sum_{i=1}^{k}\min_{\|X_{i.}-(Z-1/L\nabla f_0(Z))_{i.}\|_2\le(1-\lambda)/2L} : \|X_{i.}\|_2^2.
\end{aligned} \tag{12}$$

Therefore, the optimal solution of the first problem in (12) is equivalent to the last problem in (12). Actually, each row of X can be optimized independently in the last problem. Considering each row of X, respectively, we can get the closed form as $\min_X p_{Z,L}(X) = \mathcal{D}_{(1-\lambda)/2L}(Z - 1/L\nabla f_0(Z))$.

References

[1] R. Junee, "Zoinks! 20 hours of video uploaded every minute!," *YouTube Blog*, vol. 20, 2009.

[2] J. Luo, C. Papin, and K. Costello, "Towards extracting semantically meaningful key frames from personal video clips: From humans to computers," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 289–301, Feb. 2009.

[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[4] J. Wu, H. Christensen, and J. Rehg, "Visual place categorization: Problem, dataset, and algorithm," in *Proc. IROS*, 2009.

[5] J. Wu and J. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1489–1501, Aug. 2010.

[6] B. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Trans. Multimedia Comput., Commun., Appl. (TOMCCAP)*, vol. 3, no. 1, p. 3, 2007.

[7] A. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," *J. Vis. Commun. Image Represent.*, vol. 19, no. 2, pp. 121–143, 2008.

[8] Guidelines for the TRECVID 2007 Evaluation (2008). [Online]. Available: http://www-nlpir.nist.gov/projects/tv2007/tv2007.html, 2008

[9] B. Li and I. Sezan, "Semantic sports video analysis: Approaches and new applications," in *Proc. ICIP*, 2003, vol. 1.

[10] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. Huang, "Effective and efficient sports highlights extraction using the minimum description length criterion in selecting GMM structures (audio classification)," in *Proc. ICME*, 2004, vol. 3, pp. 1947–1950.

[11] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proc. 8th ACM Int. Conf. Multimedia*, 2000, p. 115.

[12] C. Cheng and C. 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.

[13] J. Kender and B. Yeo, "On the structure and analysis of home videos," in *Proc. ACCV*, 2000.

[14] T. Zhang, "Intelligent keyframe extraction for video printing," *Proc. SPIE*, vol. 5601, p. 25, 2004.

[15] Y. Ma, L. Lu, H. Zhang, and M. Li, "A user attention model for video summarization," in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, p. 542.

[16] A. Aner-Wolf and J. Kender, "Video summaries and cross-referencing through mosaic-based representation," *Comput. Vis. Image Understand.*, vol. 95, no. 2, pp. 201–237, 2004.

[17] Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. ICIP*, 1998, pp. 866–870.

[18] P. Gresle and T. Huang, "Gisting of video documents: A key frames selection algorithm using relative activity measure," in *Proc. 2nd Int. Conf. Visual Information Systems*, 1997, pp. 279–286.
[19] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1097–1105, Dec. 2005.
[20] T. Liu, H. Zhang, and F. Qi, "A novel video key-frame-extraction algorithm based on perceived motion energy model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 1006–1013, Oct. 2003.
[21] C. Xu, N. Maddage, and X. Shao, "Automatic music classification and summarization," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 441–450, May 2005.
[22] J. You, G. Liu, L. Sun, and H. Li, "A multiple visual models based perceptive analysis framework for multilevel video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 273–285, Mar. 2007.
[23] B. Yu, W. Ma, K. Nahrstedt, and H. Zhang, "Video summarization based on user log enhanced link analysis," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 382–391.
[24] Y. Peng and C. Ngo, "Clip-based similarity measure for query-dependent clip retrieval and video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 612–627, May 2006.
[25] Z. Li, G. Schuster, and A. Katsaggelos, "Minmax optimal video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, p. 1245, Oct. 2005.
[26] Z. Li, G. Schuster, A. Katsaggelos, and B. Gandhi, "Rate-distortion optimal video summary generation," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1550–1560, Oct. 2005.
[27] Z. Li, P. Ishwar, and J. Konrad, "Video condensation by ribbon carving," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2572–2583, Nov. 2009.
[28] Y. Fu, Y. Guo, Y. Zhu, F. Liu, C. Song, and Z. Zhou, "Multi-view video summarization," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 717–729, Nov. 2010.
[29] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc.: Series B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.
[30] K. Huang and S. Aviyente, "Sparse representation for signal classification," *Adv. Neural Inf. Process. Syst.*, vol. 19, p. 609, 2007.
[31] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Proc. 2009 IEEE 12th Int. Conf. Computer Vision*, 2009, pp. 2114–2121.
[32] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 3449–3456.
[33] Y. Nesterov, *Gradient Methods for Minimizing Composite Objective Function*. Louvain-la-Neuve, Belgium: CORE, 2007.
[34] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proc. ICML*, 2009, pp. 457–464.
[35] M. Stricker and M. Orengo, "Similarity of color images," *Proc. SPIE Storage and Retrieval for Image and Video Databases*, vol. 2420, pp. 381–392, 1995.
[36] A. Loui, J. Luo, S. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa, "Kodak's consumer video benchmark data set: Concept definition and annotation," in *Proc. Int. Workshop Multimedia Information Retrieval*, 2007, pp. 245–254.

Yang Cong (S'09–M'11) received the B.Sc. degree from Northeast University, Shenyang, China, in 2004, and the Ph.D. degree from State Key Laboratory of Robotics, Chinese Academy of Sciences, Shanghai, in 2009.

He is a Research Fellow of the National University of Singapore (NUS) and Nanyang Technological University (NTU) from 2009 to 2011, respectively. Now, he is an Associate Researcher of the Chinese Academy of Sciences. His current research interests include compute vision, pattern recognition, multimedia, and robot navigation.

Junsong Yuan (M'08) received the Ph.D. and M.Eng degrees from Northwestern University, Chicago, IL, and National University of Singapore, respectively.

Before that, he graduated from the special program for the gifted young in Huazhong University of Science and Technology, Wuhan, China. He is currently a Nanyang Assistant Professor at Nanyang Technological University, Singapore. From 2003 to 2004, he was a research scholar at the Institute for Infocomm Research, Singapore. His current research interests include computer vision, video data mining and content analysis, and multimedia search. He has filed three US patents.

Dr. Yuan was a recipient of the Doctoral Spotlight Award from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), a recipient of the elite Nanyang Assistant Professorship from Nanyang Technological University, and a recipient of the Outstanding Ph.D. Thesis award from the Electrical Engineering and Computer Science Department at Northwestern University. He currently serves as an editor of the *KSII Transactions on Internet and Information Systems*.

Jiebo Luo (S'93–M'96–SM'99–F'09) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, in 1989 and the Ph.D. degree from the University of Rochester, Rochester, NY, in 1995.

He was a Senior Principal Scientist with the Kodak Research Laboratories in Rochester before joining the Computer Science Department at the University of Rochester in Fall 2011. His research interests include signal and image processing, machine learning, computer vision, and the related multi-disciplines such as multimedia data mining, medical imaging, and computational photography. He has authored over 180 technical papers and holds over 60 U.S. patents.

Dr. Luo has been actively involved in numerous technical conferences, including serving as the general chair of ACM CIVR 2008; program co-chair of IEEE CVPR 2012, ACM Multimedia 2010, and SPIE VCIP 2007; area chair of IEEE ICASSP 2009–2011, ICIP 2008–2011, CVPR 2008, and ICCV 2011; and an organizer of ICME 2006/2008/2010 and ICIP 2002. Currently, he serves on multiple IEEE SPS Technical Committees (IMDSP, MMSP, and MLSP). He has served on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Pattern Recognition*, *Machine Vision and Applications*, and the *Journal of Electronic Imaging*. He is a Kodak Distinguished Inventor, a winner of the 2004 Eastman Innovation Award, and a Fellow of SPIE and IAPR.