Discovering Video Shot Categories by Unsupervised Stochastic Graph Partition

Xiaohua Duan, *Member, IEEE*, Liang Lin, *Member, IEEE*, and Hongyang Chao, *Member, IEEE*

***Abstract*—Video shots are often treated as the basic elements for retrieving information from videos. In recent years, video shot categorization has received increasing attention, but most of the methods involve a procedure of supervised learning, i.e., training a multi-class predictor (classifier) on the labeled data. In this paper, we study a general framework to unsupervisedly discover video shot categories. The contributions are three-fold in feature, representation, and inference: (1) A new feature is proposed to capture local information in videos, defined with small video patches (e.g., $11 \times 11 \times 5$ pixels). A dictionary of video words can be thus clustered off-line, characterizing both appearance and motion dynamics. (2) We pose the problem of categorization as an automated graph partition task, in that each graph vertex represents a video shot, and a partitioned sub-graph consisting of connected graph vertices represents a clustered category. The model of each video shot category can be analytically calculated by a projection pursuit type of learning process. (3) An MCMC-based cluster sampling algorithm, namely Swendsen-Wang cuts, is adopted to efficiently solve the graph partition. Unlike traditional graph partition techniques, this algorithm is able to explore the nearly global optimal solution and eliminate the need for good initialization. We apply our method on a wide variety of 1600 video shots collected from Internet as well as a subset of TRECVID 2010 data, and two benchmark metrics, i.e., Purity and Conditional Entropy, are adopted for evaluating performance. The experimental results demonstrate superior performance of our method over other popular state-of-the-art methods.**

***Index Terms*—Category discovery, graph partition, unsupervised categorization, video shot.**

I. Introduction

With development of multimedia technology, the number of videos on the Internet keeps increasing rapidly. *YouTube* (http://www.youtube.com/t/press_statistics) reported that more than 13 million hours of video were uploaded during 2010 and 48 hours of video are uploaded every minute. The research of managing the large amount of video data receives increasing attention. In many tasks of multimedia processing [1]–[3], video shots are often treated as the basic elements for video analysis. Following the definition in the cinema world, a video shot includes a sequence of consecutive frames captured from a certain scene, and the camera is movable in the scene.

In this paper, we study one recently arising problem in video management—automated discovering categories for a set of unlabeled video shots. Specifically, with this technology, a batch of unlabeled video shots can be automatically grouped into different categories according to their contents (i.e., appearances and motions). We can apply this approach in video database construction, similar video search, and video content analysis etc. As illustrated in Figs. 5 and 6, a few video shots sampled from our testing database exhibit various texture and structure appearance as well as motion dynamics.

A. Related Work

In the literature of multimedia and image/video processing, many efforts have been dedicated to automatically discovering categories from still images or videos.

Previous methods on unsupervised image categorization [4], [5] compute a number of low-level visual features (such as RGB color or texture) over the global domain of an image, and pool them into a feature vector, (i.e., each image is represented by a feature vector); a clustering algorithm, e.g., k-means or ISO-Data, is then performed with these feature vectors to categorize images into K groups. These methods are greatly improved by using a more effective image representation, e.g., bag-of-words [6]–[10], in which an image is represented by an orderless collection of salient words by using a visual dictionary. Based on the bag-of-words representation, the probabilistic Latent Semantic Analysis (pLSA) model was adopted to discover categories from images [11], attempting to explain the distribution of features in the image as a mixture of a few "semantic topics". As an alternative for modeling latent semantic topics, the Latent Dirichlet Allocation (LDA) model [12], [13] was widely used as well.

For the task of discovering video shot categories, which is the focus of this paper, many systems utilize the achievements in image categorization [1], [3], [14]–[16]. Exploring motion information is a non-trivial problem for video shot categorization. Ngo *et al.* [2] proposed to use statistics of motion vector for clustering and retrieval of video shots. Gupta *et al.* [17] extracted the 3D key points over scales to represent videos and performed categorization similarly with the "bag of words" based methods for classifying image. Liu *et al.* [18] proposed a multi-modality

Manuscript received July 09, 2011; revised October 30, 2011 and February 20, 2012; accepted May 07, 2012. Date of publication October 16, 2012; date of current version December 12, 2012. The work was supported by the National Natural Science Foundation of China (Grant No.61173082 and No.61173081), the Hi-Tech Research and Development Program of China (National 863 Program, Grant No.2012AA011504), the Guangdong Natural Science Foundation (Grant No.S2011020001215 and No.S2011010001378), and the Guangdong Science and Technology Program (Grant No.2010A040307003 and No.2011B040300029). This work is also supported by the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-12KF-06). The corresponding author is L. Lin. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xian-Sheng Hua.

The authors are with Sun Yat-Sen University, Guangzhou 510006, China (e-mail: linliang@ieee.org).

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.2012.2225029

video shot clustering method based on the tensor representation and Affinity Propagation algorithm [19].

By reviewing the previous work, we summarize three key problems in unsupervised categorization of video shots as follows.

1) The Effective Video Descriptors Combining Appearance and Motion Information: The classic image descriptors [9], [20]–[22], focusing on the textural or structural information of static image domains, could not be good descriptors for representing videos. For example, a video shot of one soccer game, including grass, players and sky, may be not well differentiated with an outdoor landscape shot using the static features (i.e., some video shots should be distinguished by dynamic motion information). Recently, video patch based features were proposed to capture local information in both space and time in videos [23], [24], and worked well in high-level applications, e.g., human action analysis and video object retrieval. These methods inspire us to explore good video descriptors for video shots.

2) The Adaptive Feature Selection for Different Categories: There is a clear observation that different categories of video shots might be better distinguished from each other by using different combination of features, indicating adaptive feature selection is a must. In the recent progress of computer vision, many feature selection algorithms, such as Adaboost [25], achieve state-of-the-arts performance for a number of supervised classification problems. However, most previous works of unsupervised categorization have not addressed the feature selection problem very well; instead, for each category, they use a long feature vector consisting of histograms of gradient orientations, color, and/or various filters.

3) The Efficient Clustering Algorithm for Automatic Cluster Number Determination: The inference of clustering is another key step for discovering categories and each cluster is actually a category of grouped video shots. In the previous methods, the cluster number is often prefixed or selected exhaustively in a small given range [26], [27]. The essence for automatic cluster number determination lies in whether an algorithm can explore the solution space efficiently and globally rather than exhaustive searching. In the literature, the stochastic sampling algorithms [28]–[31] were proposed to solve the clustering tasks without prefixed cluster number, such as image segmentation and perceptual grouping, but have not been adopted to discover video shot categories yet.

B. Overview of Our Method

Addressing the above mentioned problems, we propose a general method for unsupervised video shot categorization. In many previous works of classification (e.g., object classification [7] and event analysis [8]), graph representations are often adopted to represent each data entities as a graph vertex and incorporate mutual interactions among data entities as graph edges. Then the classification problem is posed as a graph partition task in general, and each partitioned sub-graph after optimization indicates a discovered category. Following these methods, we treat the video shots to be categorized as the graph vertices and solve the graph partition by a novel cluster sampling algorithm. Moreover, we combine the category model learning with the process of categorization.

We briefly introduce three core components of our method as follows.

1) Video Representation With Video Words: We propose effective features (descriptors) to represent videos, which are defined based on small video patches (e.g., $11 \times 11 \times 5$ pixels, namely "video brick"). In the literature, some efforts of manifold learning for image patches [32]–[34] reveal that (i) a small image patch is either structural (including simple and regular image primitives) or textural (including complex and stochastic patterns); (ii) These image patches can be thus represented by either a low dimensional explicit manifold or a high dimensional implicit manifold, respectively. Accordingly, Zhao *et al.* [35] studied the mathematical manifolds of video bricks recently. By analogy, we define two types of video words: explicit video words (EVWs) and implicit video words (IVWs). An EVW, a structural video word, is defined with an explicit function of one (or few) large base vectors, e.g., to represent a moving structural primitive. An IVW is described by a histograms for texture gradients or color respectively. By extracting a large number of video bricks from natural videos and quantizing them with these two definitions, we can further build up two word dictionaries including either EVWs or IVWs. Therefore, a video shot can be represented by a bag of video words using these two dictionaries.

2) Pursuit of Category Models by Information Projection: For a set of video shots grouped together, we shall adaptively select appropriate features (video words) for modeling these video shots, namely, pursuing the category model. To accomplish this task, we may use discriminative methods or generative methods. In some recent work, feature selection is performed toward a discriminative goal by optimizing the classification error (e.g., Adaboost [25]), given a set of labeled negative and positive samples. When there are unlabeled data, particularly for the unsupervised categorization problem, it is desirable to have a generative learning framework. In this work, we adopt the information projection strategy for selecting informative and discriminative features during the process of categorization, which has shown promising results on generative learning under an information-theoretic framework [32], [33], [36]. As a result, the generative model for each category can be pursued by maximizing information gain of selected features. It is worth mentioning that the model pursuing is performed simultaneously together with the clustering procedure.

3) Stochastic Cluster Sampling for Graph Partition: The computation of clustering is the key component, which is posed as a graph partition task via the graph representation in which each vertex specifying a video shot. The graph partition for our task is challenging due to two characters: the unknown number of underlying categories and no confident initialization. We formulate the graph partition with a probabilistic form that the posterior probability of graph partition is defined as the accumulation of the generative models of all categories plus the weak priors. Intuitively, the goodness of partition is determined based on how well the pursued models explain or represent the partitioned categories. And the number of categories, (i.e., the partition number), can be inferred together with the graph parti-

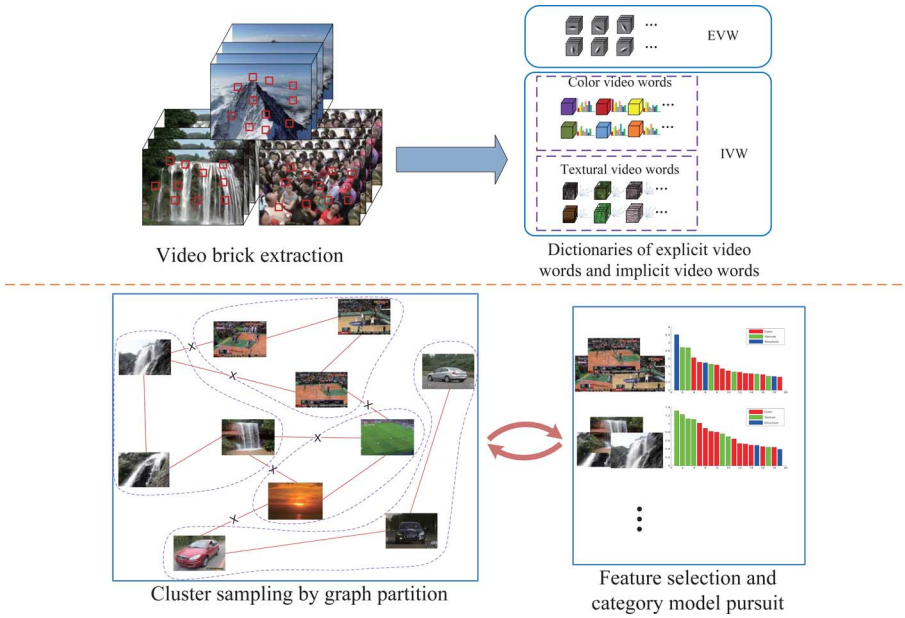

Fig. 1. Overview of the proposed method. As the upper panel illustrates, we extract a number of video bricks and construct the dictionaries of video words (i.e., explicit video words and implicit video words). In the lower panel, we illustrate the procedure of discovering categories of video shots; it is posed as a unsupervised graph partition task. Moreover, we simultaneously learn the probability model for each category, as the informative video words are selected.

tion inference. Therefore, solving the optimal graph partition is equivalent to maximizing the posterior probability under the Bayesian framework. The optimized partition as well as the number of category is approximate to the groundtruth. In the literature, there are many effective algorithms for solving graph partition. Some deterministic algorithms, such as graph-cuts and belief propagation [37], are very fast but need a good initialization; some stochastic algorithms, e.g., Gibbs Sampling[29], can proceed without a good initialization but they are limited by the low efficiency. In our method, we utilize a recently proposed cluster sampling algorithm, namely Swendsen-Wang cuts (SWC) [28], and make improvement for its efficiency. The algorithm is able to fast partition the graph into an unknown number of clusters with a random initialization [28]. In each sampling step, the algorithm stochastically explores new partition solutions by implementing reversible jumps, (i.e., creating a new partition or merging two partitions), and the acceptance rate of the new solution is decided by the Metropolis-Hastings mechanism [31].

The framework of our approach is summarized in Fig. 1, which includes two stages. In the first stage, as shown in the upper panel in Fig. 1, two dictionaries of video words (i.e., EVWs and IVWs) are constructed. The EVWs are defined with a bank of spatio-temporal primitives, i.e., moving Gabor elements in space and time. The IVWs are obtained from a collection of natural videos. A number of small video patches, namely video bricks, are randomly extracted from videos and vectorized with the color histogram and the texture gradient histogram respectively; we then cluster these feature vectors into a small number of words, color video words and textural video words, by the k-means clustering algorithm. In the second stage of our method, we categorize the video shots by unsupervised graph partition. Each video shot is represented with the dictionaries of EVWs and IVWs. The graph partition is solved by a Swendsen-Wang cuts sampling algorithm. The partition sampling iterates simultaneously together with the pursuit of category models, as illustrated in the down panel in Fig. 1.

The key contributions of this paper are as follows. First, we propose a general approach for discovering video shot categories with stochastic cluster sampling via graph representation. Second, two types of video words are proposed to well capture local appearance and motion information in video shots. Third, the generative category models are pursued simultaneously together with the clustering procedure, using an information gain criterion. Last, we adopt a cluster sampling algorithm for efficient inference and the number of categories is automatically determined. Our method is evaluated on two public datasets and it outperforms the state-of-the-arts approaches.

The rest of this paper is organized as follows. We first introduce the spatio-temporal video words in Section II. Then we present problem formulation of discovering video shot categories under the Bayesian framework in Section III, and follow with an description of the category model pursuit in Section IV. Section V presents the cluster sampling algorithm for inference. The experimental results and comparisons are exhibited in Section VI, and the paper is concluded in Section VII.

II. Video Words: Integrating Appearance and Motion Information

In this section, we propose two types of brick-like video words integrating appearance and motion information, so that a video shot can be thus represented by a bag-of-words model.

A. Explicit and Implicit Video Words

In recent literature, spatio-temporal video patches have been proposed in applications of video segmentation [38], object tracking [39], [40], and action recognition [41], [42]. Inspired by these works, we define the brick-like video words in space

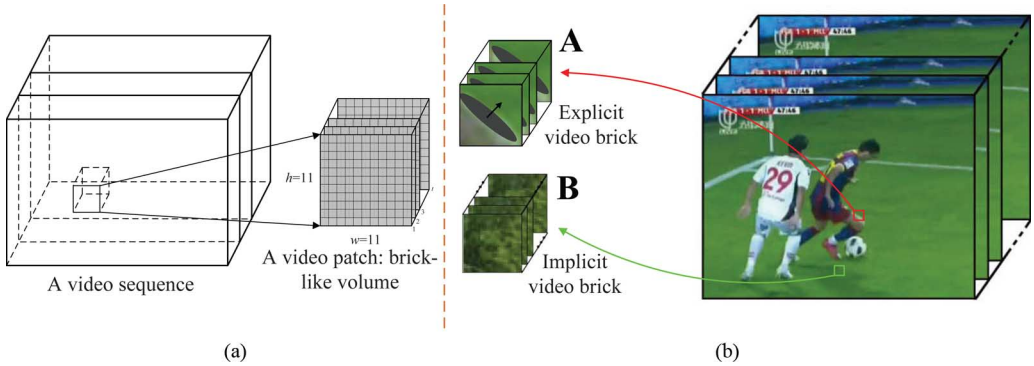

Fig. 2. (a) A video patch is a brick-like volume defined in space and time. (b) According to the appearance and motion information, we explore two ways, explicit and implicit, for characterizing video bricks.

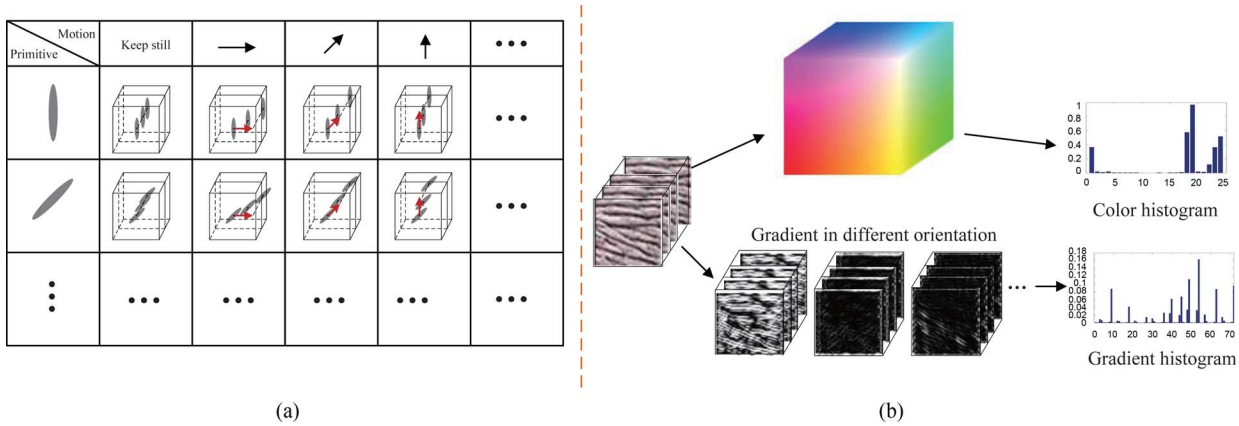

Fig. 3. Illustration of explicit video words and implicit video words. An explicit video word (EVW) is defined with moving Gabor elements (denoted as an ellipse), which can move along with 8 possible directions or keep still, as shown in (a). An implicit video word (IVW) is defined with a statistical histogram. In (b), given a video brick, we can map it into two feature spaces for calculating the histogram: (i) discretized colors histogram, and (ii) histogram of orientated gradients. See the text for video word definition in details.

and time. We regard a video shot as a 3D structure consisting of video patches, as shown in Fig. 2(a). We fix the small size of a video brick as 11×11 pixels in spatial domain, and 5 frames in time, which is less affected by compositionality and thus remains "pure". That is, these video bricks of small size include relative simple content and can be thus described by one single type of feature. Reversely, video bricks of large size (e.g., $30 \times 30 \times 20$ pixels) are probably complex due to comprising objects with various appearances and motions, so that some hybrid or mixed features are often employed.

To capture and represent the information of such a video brick (in the size of $11 \times 11 \times 5$ pixels), we consider both appearance and motion properties. There are two types of video bricks, which can be characterized in two different ways. The video shot of a soccer game (Fig. 2(b)) is a good example. One video brick from the player's leg can be clearly defined with a moving Gabor element (denoted by a eclipse). The response of the Gabor captures the appearance and the motion vector (denoted by an arrow) of the Gabor represents the motion dynamic. Thus, we refer this type of video bricks as "explicit". By contrast, another video brick from the grass has no explicit property in both appearance and motion, and thus can be only characterized by the implicit statistics, such as color or textural histograms. Since video bricks are fixed in small size, we assume that each of them can be described by explicitly or implicitly.

With the above observation, we further propose two type of video words, namely the Explicit Video Words (EVWs) and Implicit Video Words (IVWs), to represent a video shot. In the perspective of mathematics, a video word is an ensemble or equivalence class of the video brick instances that share the same definition. We refer a video brick by $\mathbf{B}$, and propose the definitions as follows.

Definition 1: An explicit video word (EVW) is a cluster (small set) of bricks spanned by one basis function,

$$\omega^e = \{\mathbf{B} : \mathbf{B} = cG_i + \varepsilon\},$$

where G_i is a primitive function selected from a family of spatio-temporal basis $\Delta_e = \{G_i\}_{i=1}^N$, c is the reconstruction coefficients and ε is the residual.

As shown in Fig. 3(a), we define the spatio-temporal basis Δ_e by moving Gabor wavelets. We use large scale 2D Gabor wavelets [43] at 36 orientations at size of 11×11 pixels; they move in 8 directions with step of 2 pixels or keep still. So we can obtain $N = 36 \times 9 = 324$ spatio-temporal primitives, each of which represents an explicit video word in the dictionary.

Definition 2: An implicit video word (IVW) is a cluster (small set) of bricks that share a similar statistic,

$$\omega^m = \left\{\mathbf{B} : H(\mathbf{B}) = \hat{H} + \varepsilon\right\},$$

where $H(\mathbf{B})$ denotes the texture or color histogram over the video brick $\mathbf{B}$, $\hat{H}$ is the mean histogram of the bricks, and ε is the statistical fluctuation, i.e., a very small value.

As shown in Fig. 3(b), to well explore both color and texture information, we further define two types of implicit video words: textural video words (TVW) and color video words (CVW), which have the same form but different types of histogram.

For defining TVWs, we adopt the orientated gradient histogram $H(\mathbf{B}) = (h_1, h_2, \ldots, h_N)$, and each bin h_i denotes a quantized value of orientated gradients. We discrete the gradients into 8 orientations in both spatial and temporal domains, thus obtaining a histogram of dimension $N = 8 \times 8 = 64$. For each pixel in a video brick, its gradients in both spatial and temporal domains are calculated and pooled into the histogram. In constructing TVWs, we first collect many textural video bricks from our dataset and calculate the histogram for each, and then group them into a number of clusters using the k-means algorithm. Each cluster represents a TVW and $\hat{H}$ is calculated by averaging the histograms in the cluster.

The CVWs are constructed exactly the same as the TVWs but using the color histogram. In our implementation, we use a 24-dimension histogram in HSV space (18 bins for hue, 3 bins for saturation, and 3 bins for value).

Given a video word dictionary $\mathbb{W} = \{\omega_i,\ i = 1, \ldots, M\}$, where ω_i is a video word (EVW, TVW or CVW), any video shot v can be represented as a vector $\mathbf{R} = (R_1(v), R_2(v), \ldots, R_M(v))$, where $R_i(v)$ is the response with the video word ω_i, and

$$R_i(v) = h\left(\sum_{\mathbf{B}\in v} \mathbf{1}_{\omega_i}(\mathbf{B})\right), \tag{1}$$

where $\mathbf{1}_{\omega_i}(\mathbf{B})$ is the indicator function to indicate whether the video brick $\mathbf{B} \in v$ matches with the word ω_i, i.e., $\mathbf{1}_{\omega_i}(\mathbf{B}) = \{1|0\}$. Thus, $\sum_{\mathbf{B}\in v} \mathbf{1}_{\omega_i}(\mathbf{B})$ intuitively indicates the number of the video word ω_i appearing in the video shot v. $h(\cdot)$ is the sigmoid transformation which is characterized by a saturation level ξ (e.g., $\xi = 6$),

$$h(x) = \xi\left(\frac{2}{1 + e^{-2x/\xi}} - 1\right), \tag{2}$$

which increases from 0 to ξ.

III. Problem Formulation

Given a batch of unlabeled video shots $\mathbb{D}$ and a dictionary of video words $\mathbb{W}$, we discover their categories by partitioning them into an unknown number of disjoint K groups, as

$$\Pi = \{D_1, D_2, \ldots, D_K\},\ \cup_{k=1}^{K} D_k = \mathbb{D},\ D_i \cap D_j = \emptyset,\ \forall i \neq j. \tag{3}$$

The partition of video shots is defined in an adjacency graph $G_0 = \langle V, E_0\rangle$, in which $V = \mathbb{D} = \{v_1, v_2, \ldots, v_N\}$ is the set of graph vertices specifying the video shots to be categorized, and E_0 is the set of edges connecting neighboring graph vertices. We solve the task of graph partition by cutting edges, i.e., generating disjoint subgraphs. However, G_0 is a fully connected graph where the initial edge set E_0 could be very large, leading to intractable inference. We thus need to compute a relatively sparse graph representation $G_0 = \langle V, E\rangle$ by pruning edges, $E \subset E_0$ at the initialization step, which will be introduced in Section V.

In our method, the graph partition is solved iteratively together with learning the probability models for each category D_k, as,

$$\Psi = \{P_k(W_k, \Theta_k),\ W_k \subset \mathbb{W},\ k = 1, \ldots, K\}, \tag{4}$$

where W_k denotes the selected video words for modeling the category D_k and Θ_k includes the corresponding model parameters, i.e., the coefficients of words.

Thus, we define the solution representation as

$$S = \big(K, \Pi, \Psi\big), \tag{5}$$

where K denotes the category number, Π and Ψ denote the category partitions and category models respectively, and they are mutually conditional and closely coupled. Given a state of partition, we can learn (or update) the probability models while the category models can drive the partition to be refined. Compared with the previous unsupervised categorization approaches [27], we integrate the category number K into the solution S, indicating that K should be solved together with category partitioning and model learning.

We seek the optimal S^* by maximizing a posterior probability (MAP),

$$S^* = \arg\max_{S\in\Omega} p(S|\mathbb{D}), \tag{6}$$

where Ω is the solution space for inference. We factorize the posterior probability under the Bayesian framework,

$$p(S|\mathbb{D}) \propto p(S)p(\mathbb{D}|S), \tag{7}$$

where $p(S)$ and $p(\mathbb{D}|S)$ denote the prior model and likelihood model respectively. Without loss of the generality, we define the prior model by incorporating an exponential function for K and an uniform function for Π and Ψ. The likelihood model $p(\mathbb{D}|S) = p(\mathbb{D}|\Pi, \Psi)$ can be defined as a product of generative models of all separated categories. Assume each category D_k includes n_k video shots (i.e., graph vertices), we have

$$\begin{aligned} p(\mathbb{D}|\Pi, \Psi) &= \prod_{k=1}^{K} P_k(W_k, \Theta_k) \\ &= \prod_{k=1}^{K}\prod_{i=1}^{n_k} P(v_{k,i}, W_k, \Theta_k). \end{aligned} \tag{8}$$

Therefore, the posterior probability can be further written as,

$$p(S|\mathbb{D}) \propto \exp\{-\beta K\} \prod_{k=1}^{K}\prod_{i=1}^{n_k} P(v_{k,i}, W_k, \Theta_k), \tag{9}$$

where β is an empirical parameter constraining the number of inferred categories. The probability model $P_k(W_k, \Theta_k)$ for each category D_k can be learned and updated by a simple yet effective generative learning algorithm, which will be introduced in the next section.

IV. Learning Generative Category Model

Given a graph partition Π, we shall learn the probability model $P_k(W_k, \Theta_k)$ for each category D_k, which is equivalent to selecting features (i.e., video words) $W_k \subset \mathbb{W}$ to describe D_k.

Since all video shots from $\mathbb{D}$ are unlabeled without extra negative samples, we adopt an efficient generative learning algorithm, namely information projection [32], [33], to select features according to the information gain criterion.

Suppose the category D_k is governed by an underlying target model F_k, the model pursuit can be viewed as finding a sequence of informative features from an initial model $P_{k,0}$. At each step t, the model $P_{k,t}$ is updated to gradually approach F_k,

$$P_0 \rightarrow P_1 \rightarrow \ldots \rightarrow P_t \quad to \quad F, \tag{10}$$

in terms of minimizing the Kullback-Leibler divergence $\mathcal{K}(F\|P_t)$, where we neglect k for simplicity. In the manner of stepwise pursuing, the new model P_t is updated by adding a new feature based on the current model P_{t-1}.

$$P_t^* = \arg\min \mathcal{K}(P_t\|P_{t-1}), \tag{11}$$

subject to a new constraint with regard to the new feature ω_t,

$$E_{P_t}[R_t] = E_F[R_t], \tag{12}$$

where R_t denotes the response of the word ω_t. Intuitively, we search the optimal new model P_t^* closest to current model P_{t-1}, minimizing $\mathcal{K}(P_t\|P_{t-1})$, because the previous constraints in P_{t-1} should be preserved. $E_F(R_t)$ represents the expectation of feature ω_t over the underlying model, i.e., the marginal distribution projected into the feature ω_t, which can be calculated by averaging feature responses over positive samples. $E_{P_t}[R_t]$ denotes the feature expectation on updated model.

By solving this constrained optimization problem by Lagrange multiplier in (11), we have

$$P_t(v) = \frac{1}{z_t} P_{t-1}(v) e^{\lambda_t R_t(v)}, \tag{13}$$

λ_t is the coefficient weight of the selected feature ω_t and z_t normalizes the probability to 1. The probability model can be obtained by T rounds of feature selection,

$$P(v;\Theta) = P_0(v)\frac{1}{Z}\exp\left\{\sum_{t=1}^{T}\lambda_t R_t(v)\right\}. \tag{14}$$

$Z = \prod z_t$ and $\Theta = (\lambda_1, \ldots, \lambda_T)$. In our implementation, in order to make different category model comparable, we use the same reference model for P_0 (which could be the uniform distribution or Gaussian white noise distribution). In the original version of information projection [36], the probability model $P(v;\Theta)$ is computational expensive, since in each step t, it needs to draw samples for synthesizing current distribution from the model P_{t-1} for calculating $\mathcal{K}(P_t\|P_{t-1})$. Based on the recently proposed improvement [32], [33], if we enforce that the selected features have little overlap in both spatial and frequency domain, we may simply assume that $P_{t-1} = P_0$ as an approximation, and $\mathcal{K}(P_t\|P_{t-1}) \approx \mathcal{K}(P_t\|P_0)$ in (11). Intuitively, all features are selected independently. Thus, the feature weight λ_t can be estimated by MLE over positive samples (i.e., video shots in category D_k),

$$L(\lambda_r) = \frac{1}{n_k}\sum_{i=1}^{n_k} R_t(v_i), \tag{15}$$

where n_k is the number of video shots in category D_k. The normalization term for each round z_t can be also solved accordingly, $z_t = E_{P_0}(\exp\{\lambda_t R_t(v)\})$. Furthermore, the probability model can be factorized as

$$P(v;\Theta) = P_0(v)\prod_t^T\left[\frac{1}{z_t}\exp\{\lambda_t R_t(v)\}\right]. \tag{16}$$

This model can be learned by selecting the most informative feature ω_t^* having maximum information gain at each round t, denoted by $\Delta(\omega_t)$, to the current model P_{t-1},

$$\omega_t^* = \arg\max \Delta(\omega_t), \tag{17}$$

$$\Delta(\omega_t) = \mathcal{K}(P_t\|P_{t-1}) = \lambda_t R_t(v) - \log z_t. \tag{18}$$

The feature responses $R_i(v_j)$, $R_i \sim \omega_i \in \mathbb{W}$, $v_j \in \mathbb{D}$ over all video shots can be calculated off-line, as well as the information gain $\Delta(\omega_i)$, which makes it very fast for the probability model learning. The efficiency of learning allows us to keep the category models updating during the process of discovering categories.

In our method, the inference of discovering categories, includes a series of steps, which iteratively updates (re-learns) category models and then re-groups the video shots into categories guided by the category models. We will discuss it in the next section.

V. Inference by Stochastic Sampling

Given the posterior probability in (9), we employ a stochastic cluster sampling algorithm for searching optimized solution S^*, including the graph partitions Π and corresponding category models Ψ.

$$S^* \sim p(S|\mathbb{D}). \tag{19}$$

The reasons of using stochastic scheme rather than the deterministic algorithms are as follows. (1) There is not a reliable initialization for categorization; (2) the posterior probability model is in a non-convex form so that it is unsuitable for fast gradient descent optimization; (3) it is difficult to design the heuristic optimization rules due to the unpredictable variance and ambiguity of video shots.

Cluster sampling is a powerful stochastic algorithm, designed under the Metropolis-Hasting mechanism [31]. It was first proposed by Swendsen and Wang [44] to simulate Ising/Potts graphical models in physics during the 1980s, and extended for graph partitioning by Barbu and Zhu [28], who designed the algorithm called Swendsen-Wang cuts (SWC). It is able to fast sample the optimal solution S^* for the posterior probability $p(S|\mathbb{D})$ by simulating a Markov chain which visits a sequence of states in the solution space Ω. The algorithm searches the optimal solution in the Markov chain by realizing a reversible

jump between any two successive states. We refer to [28] for the technical background.

Before introducing the inference algorithm, we first discuss the graph initialization implemented off-line.

A. Graph Initialization

According to the above explanation, the initial adjacency graph $G_0 = \langle V, E_0 \rangle$ is too complex to be inferred with, where V is the set of graph vertices specifying the video shots and E_0 is the set of edges connecting neighboring graph vertices. We need to firstly obtain a relatively sparse set of connecting edges $E \subset E_0$.

For any edge $e \in E_0$, we introduce an auxiliary random variable $\mu_e = \{\text{on}|\text{off}\}$, namely connecting variable, which indicates whether the edge is turned on or off, and the edge turn-on probability q_e is defined according to the similarity of two connected video shots,

$$q_e = p(\mu_e = on|v_s, v_t), \tag{20}$$

where v_s and v_t are two graph vertices connected by the edge e. In our method, we use the dictionary of video words $\mathbb{W}$ to measure the similarity of two arbitrary video shots. For any vertices $v \in V$, we represent them with the video words, $\mathbf{R}(v_i) = (R_1(v), R_2(v), \ldots, R_M(v))$, where $R_i(v)$ is the response with the video word ω_i, as in (1). Thus, we can fast compute the turn-on probability q_e for two arbitrary video shots $v_s \in V$, $v_t \in V$ as

$$q_e(s,t) = \exp\left\{ -\frac{\tau}{2}\left[\mathcal{K}(\mathbf{R}_s \| \mathbf{R}_t) + \mathcal{K}(\mathbf{R}_t \| \mathbf{R}_s)\right]\right\}, \tag{21}$$

where we denote $\mathbf{R}_s = \mathbf{R}(v_s)$ and $\mathbf{R}_t = \mathbf{R}(v_t)$ for notation simplicity. $\mathcal{K}$ is the Kullback-Leibler divergence for measuring two feature vectors. τ is a constant parameter. In practice, $q_e(s,t)$ should be close to 0 if v_s and v_t naturally belong to different categories, and then the edge e connecting v_s and v_t could be turned off with high probability.

For arbitrary edge $e \in E_0$ in the initial adjacency graph, we first compute the turn-on probability q_e off line. The edges with low turn-on probability can be then removed deterministically,

$$q_e = 0, \quad \text{if} \quad q_e < \eta, \tag{22}$$

where η is a tuning threshold for controlling the number of edges in the graph. Therefore, we obtain the more sparse graph $G = \langle V, E \rangle$ where $E \subset E_0$.

B. Cluster Sampling

In the following, we introduce the SWC algorithm for optimized solution inference. In general, this algorithm iterates in two steps:

Step 1: We generate the connected components (CPs) by probabilistically turning off connecting edges in the graph. Graph vertices connected together by "on" edges form a connected component (denoted by CP for simplicity). For arbitrary edge $e \in E$, we sample the connecting variable μ_e following the Bernoulli probability,

$$\mu_e \sim Bernoulli(q_e). \tag{23}$$

Then we obtain a few CPs, each of which is a set of connected graph vertices.

Step 2: We probabilistically relabel these CPs for exploring a new partition solution. In the original SWC algorithm [28], given the generated CPs, only one of them shall be selected for further relabeling. We first introduce the relabeling of the traditional SWC algorithm, and then discuss our improvement.

Assuming that the current partition solution is S_A, we are exploring a new solution S_B. We can implement two types of reversible jumps between the two state S_A and S_B by relabeling the selected CP.

- **Split**: the CP receives a new label that indicates a new category is created.
- **Merge**: the CP receives a label the same with an existing category, and then the video shots in the CP are merged into the category.

We design the relabeling by the Metropolis-Hastings method [31]. Let $Q(S_A \rightarrow S_B)$ be the proposal probability for moving from state S_A to state S_B, and conversely, $Q(S_B \rightarrow S_A)$ is the proposal probability from S_B to S_A. The acceptance rate of the move from S_A to S_B is,

$$\alpha(S_A \rightarrow S_B) = \min\left(1, \frac{Q(S_B \rightarrow S_A)}{Q(S_A \rightarrow S_B)} \cdot \frac{p(S_B|\mathbb{D})}{p(S_A|\mathbb{D})}\right). \tag{24}$$

Since given a selected CP we randomly (i.e., uniformly) perform the two types of reversible jumps for relabeling, we can simplify the ratio of the proposal probability as,

$$\frac{Q(S_B \rightarrow S_A)}{Q(S_A \rightarrow S_B)} = \frac{\prod_{e \in C_B}(1-q_e)}{\prod_{e \in C_A}(1-q_e)}, \tag{25}$$

where C_A denotes the edge set of edges that are probabilistically turned off for generating the CP on state S_A, and similarly C_B is the turning-off edge set on S_B. We intuitively call C_A or C_B as a "cut".

In such a Markov chain transition, the computation cost for each move is relatively low since the computation of the posterior probability ratio $p(S_B|\mathbb{D})/p(S_A|\mathbb{D})$ only involves the relabeling of video shots in the selected CP. Intuitively, we only need to update the models for the categories where we add or remove video shots.

In our approach, for enhancing the inference efficiency, we further improve the SWC algorithm by combining the generated CPs into compositional connected components. We thus call this algorithm compositional SWC. The benefit of this improvement is illustrated in the experiment in Section VI.

We demonstrate the key step of the compositional SWC in Fig. 4, where the different colors indicate different categories. Given a current solution state S_A (in Fig. 4(a)), we first generate a number m of CPs by turning off a few edges, as Fig. 4(b) illustrates. Then we build up a higher layer of graph $\mathbf{G}$, as shown in Fig. 4(c), in which each vertex represents one generated CP and any two neighboring CPs are connected by one edge. Being similar with the original graph G, the turn-on probability q^{CP} for each edge in $\mathbf{G}$ is calculated according to the consistency of two connecting CPs. More specifically, given the neighboring CP_i and CP_j, we define q^{CP} by integrating all the turn-on probabilities of the edges between the two CPs,

$$q^{CP} \propto \left[1 - \prod(1-q_e)\right],\ e = \langle s,t \rangle,\ s \in CP_i,\ t \in CP_j. \tag{26}$$

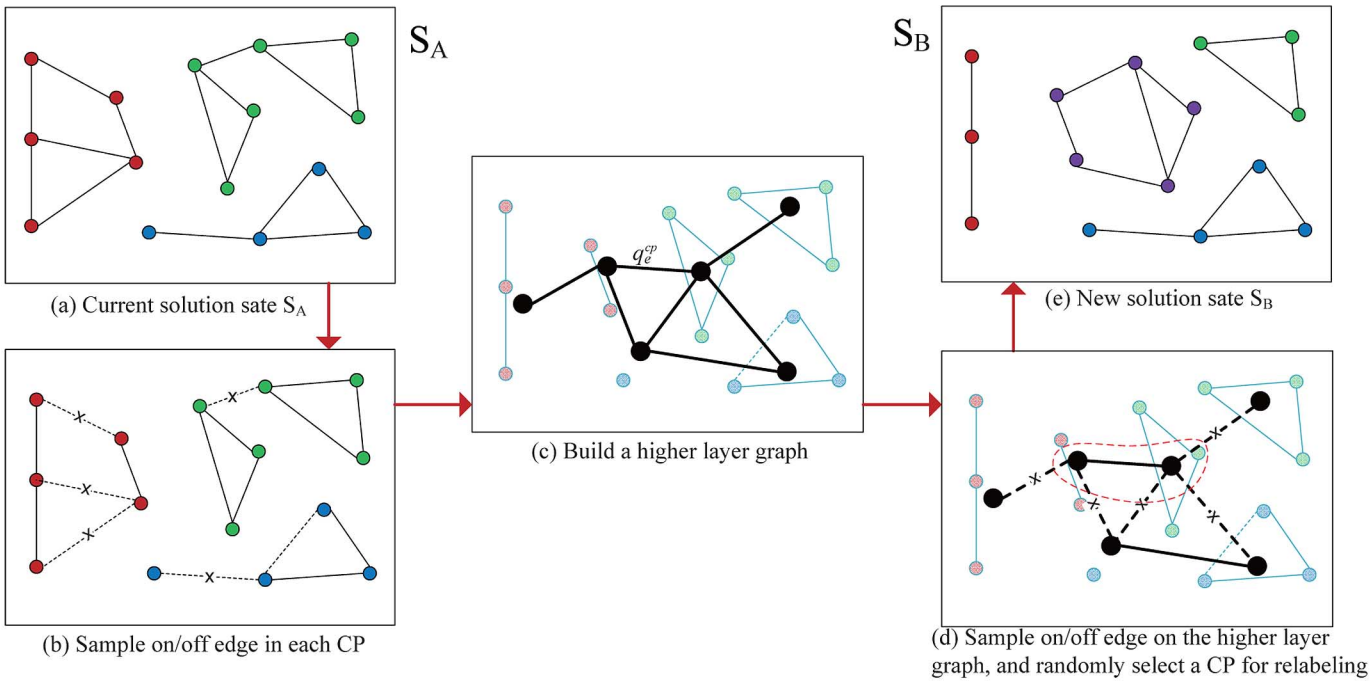

Fig. 4. Illustration of the compositional Swendsen-Wang cuts algorithm for exploring a new solution state. (a) Current solution sate S_A. (b) Sample on/off edge in each CP. (c) Build a higher layer graph. (d) Sample on/off edge on the higher layer graph, and randomly select a CP for relabeling. (e) New solution sate S_B.

In the following, we probabilistically turn off the edges in the higher layer graph $\mathbf{G}$, just like generating CPs in the original graph G, and then obtain a few groups of connected CPs, i.e., the compositional CPs in the higher layer. In Fig. 4(d), 4 compositional CPs are created, and two of them are formed by two original CPs. We can randomly select on compositional CP and relabel it with the two abovementioned reversible jumps. We enforce the compositional CP being relabeled as a complete unit, i.e., all original CPs in the compositional CP will receive the same label. As shown in Fig. 4(d), the selected compositional CP is highlighted by the red dashed circle and a new solution state S_B is achieved as shown in Fig. 4(e).

We can see that the compositional SWC algorithm enlarges the scope of the sampling algorithm. As Fig. 4 illustrates, go from S_A to S_B the traditional SWC needs at least three steps, among which there may be local minimums, whereas for the compositional SWC there is only one step.

The inference can be stopped according to the current posterior probability $p(S|\mathbb{D})$ that is kept updating during the sampling process. In practice, we define the target energy $-\log p(S|\mathbb{D})$ for better manipulation, that is, we stop the algorithm once the target energy is converged into a very small value. In addition, we design an experiment to illustrate the convergence of inference and compare with the original SWC algorithm.

Algorithm 1 summarizes our method of discovering video shot categories.

Algorithm 1: The Sketch of Discovering Shot Categories by Unsupervised Graph Partition

1. **Input**: Video shot dataset $\mathbb{D} = \{v_1, \ldots, v_N\}$, and video words $\mathbb{W} = \{\omega_i, \ldots, \omega_M\}$
2. **Output** The categorization solution $S = \left(K, \Pi, \Psi\right)$
3. Initialization;
   (1) Represent each video shot v_i with the video words, $\mathbf{R}(v_i) = \{R_1(v_i), \ldots, R_M(v_i)\}$.
   (2) Create the adjacency graph $G_0 = \langle V, E_0 \rangle$, and compute the turn-on probability q_e according to (21), $\forall e \in E_0$.
   (3) Remove the edges with low turn-on probability deterministically, as in (22), and obtain the sparse graph $G = \langle V, E \rangle$.
4. Loop for cluster sampling
   (1) At the current solution S_A, generate the CPs by probabilistically turning off connecting edges in the graph G.
   (2) Randomly select a number m of CPs for constructing a high layer of graph $\mathbf{G}$.
   (3) Generate the compositional CPs by probabilistically turning off edges in $\mathbf{G}$.
   (4) Randomly select one compositional CP and relabel it to achieve one new solution S_B.
   (5) Accept the new solution according to the acceptance rate defined in (24).
   (6) Continue the loop until the target energy $-\log p(S|\mathbb{D})$ converge into a small value.
5. Output the final solution $S^* = \arg\max p(S|\mathbb{D})$.

VI. Experiments

In this section, we evaluate the proposed method to discover categories from a number of unlabeled video shots, and compare with other state-of-the-arts approaches. The experiments

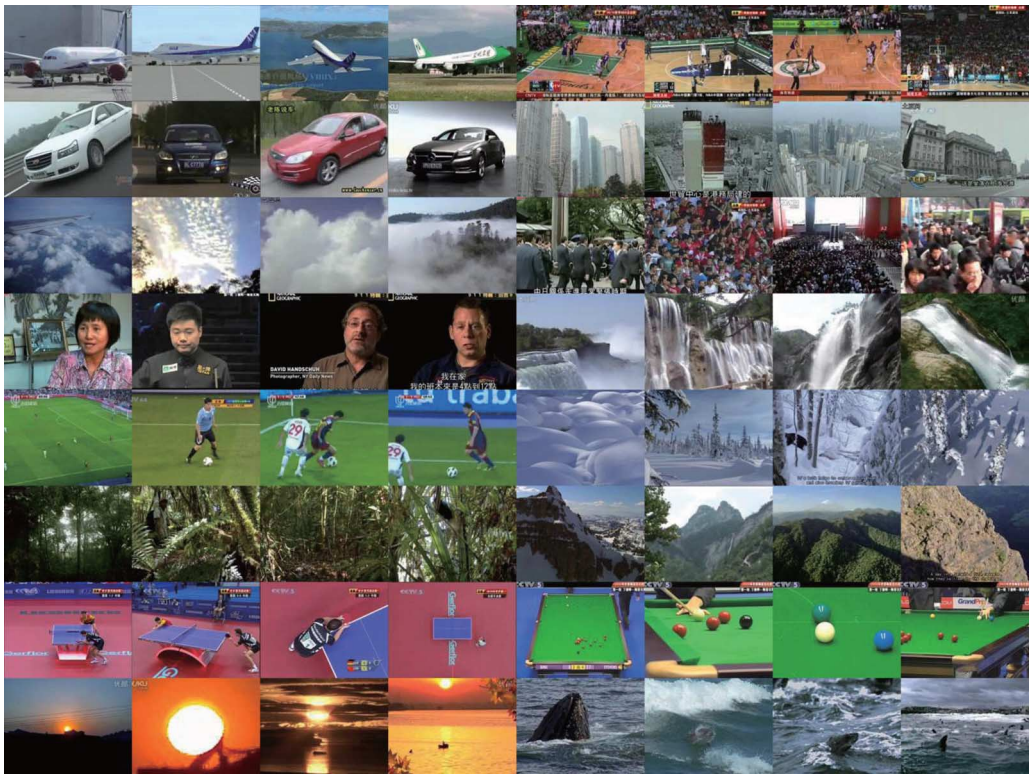

Fig. 5. Representative video shot samples from the SYSU_VIDEO_SHOTS dataset.

are carried out on two public datasets of video shots: one collected from Internet called SYSU_VIDEO_SHOTS[1] and the other selected from the TRECVID database [45]. In general, each video shot in these two datasets spans $3 \sim 60$ seconds with frame rate 25 fps.

A. Datasets and Benchmark

The SYSU_VIDEO_SHOTS dataset include 1600 video shots in total, and were classified manually into 16 categories according to their meaningful semantics. The detailed statistics of video shots are summarized in Table I, and some representative samples are shown in Fig. 5. Compared with the current public databases of video shots, the SYSU_VIDEO_SHOTS dataset has the following characters. First, the data is very compact since all video shots are carefully reviewed and some redundant entities have been removed. Second, the video shots exhibit large appearance variance as well as motion dynamics so that the dataset can be suitable for evaluating the adaptability of categorization methods (as Fig. 8 illustrates). For example, some video shots include structural moving or static objects, such as cars, faces, and airplane; some include objects of rich textural appearance and/or motion dynamics, such as cityscape and crowd. Third, the semantic topics of this dataset cover a wide range of daily videos, and some special and interesting categories are prepared, such as cloud and waterfall.

The dataset from TRECVID 2010 includes more than 8000 video shots of 40 semantic categories. Although the complete TRECVID 2010 database includes 130 semantic concepts in total, 50 of them have been annotated for evaluating classification. And 10 categories out of the 50 annotated categories include only a very small number of video shots, e.g., less than 20. Thus, we test our approach with the rest 40 annotated categories. The 40 categories are *Adult, Airplane_Flying, Animal, Asian_People, Boat_Ship, Building, Bus, Cityscape, Computer_Or_Television_Screens, Computers, Dancing, Dark-skinned_People, Demonstration_Or_Protest, Doorway, Explosion_Fire, Female_Person, Female_Human_Face_Closeup, Flowers, Ground_Vehicles, Hand, Indoor, Indoor_Sports_Venue, Infants, Instrumental_Musician, Landscape, Male_Person, Mountain, News_Studio, Nighttime, Old_People, Plant, Road, Running, Scene_Text, Singing, Telephones, Vehicle, Walking, Walking_Running, Waterscape_Waterfront,* respectively. Some representative samples are illustrated in Fig. 6.

The TRECVID database was originally proposed for evaluating semantic indexing and supervised categorization, and the

TABLE I
SYSU_VIDEO_SHOTS DATASETS FOR PERFORMANCE TESTING (#: NUMBER OF SHOTS)

Category	#	Category	#
Basketball	106	Airplane	93
Car	104	City	92
Crowd	108	Cloud	99
Waterfall	104	Face	102
Ice	99	Soccer	104
Mountain	94	Jungle	92
Snooker	102	Ping-Pong	108
Waterscape	93	Sunrise	100

[1]This dataset is now available on-line: http://gitl.sysu.edu.cn/data_videoshots.html

Fig. 6. Representative video shot samples for the dataset selected from TRECVID 2010.

evaluation metrics it compromised are not for unsupervised category discovering. More specifically, the metrics in TRECVID such as *Retrieval Rate*, *Recall* and *Precision* are defined for evaluating the performance of ranking with matching (detection) score that can be output by a retrieval or detection system; in addition, the number of categories is assumed to be fixed.

Therefore, we adopt the *Purity* and the *Conditional Entropy* as the benchmark metrics for quantitative evaluation, following the empirical survey of unsupervised discovery [27]. We briefly introduce these two metrics as follows.

For the input set $\mathbb{D}$, including a number of N shot instances, suppose the underlying category number is L and the corresponding groundtruth category labels are denoted by $X = \{x_i \in [1, L],\ i = 1, \ldots, N\}$. A system partitions the video shots into K categories, $\{D_k,\ k = 1, \ldots, K\}$, together with the inferred category labels $Y = \{y_i \in [1, K],\ i = 1, \ldots, N\}$. Note that there is a possibility of $K \neq L$ due to the automatic cluster number determination. Then the metric *Purity* and *Conditional Entropy* are, respectively, defined as,

$$Purity(X|Y) = \sum_{y \in Y} p(y) \max_{x \in X} p(x|y), \tag{27}$$

$$H(X|Y) = \sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log \frac{1}{p(x|y)}, \tag{28}$$

where $p(y) = |D_y|/N$ and $p(x|y)$ can be simply estimated from the observed frequencies in categorized data, resulting in an empirical estimation. $|D_y|$ represents the number of video shots in one category. Intuitively, the larger value of *Purity* implies the better performance in categorization and the *Conditional Entropy* is the other way round.

B. Parameters Setting and Results

We carry out the experiments on a PC with Core Duo 3.0 GHZ CPU and 16 GB memory. In the experiments, we first randomly select 1000 video shots for video brick extraction and word dictionary construction. We set the parameter β in the probabilistic formulation (9) $\beta = 600$, and the temperature parameter τ in the probabilistic edge definition (21) $\tau = 0.25$.

Since we adopt the stochastic sampling algorithm for clustering inference, we carry out our method 10 times and use the average performance for comparison. The cluster numbers inferred by our method may not be identical each time. The inferred numbers of clusters each time are reported in Table II. For SYSU_VIDEO_SHOTS dataset, the expected cluster number based on the average of 10 times is 17.6; for TRECVID data, the expected cluster number is 39.5.

TABLE II
THE INFERRED CLUSTER NUMBER IN EACH TIME OF EXPERIMENT

#	1	2	3	4	5	6	7	8	9	10
I	18	19	18	18	19	16	17	17	18	16
II	39	41	38	38	41	41	41	38	37	40

#: No. of experiments;
I: Experiments on SYSU_VIDEO_SHOTS dataset;
II: Experiments on TRECVID dataset.

For comparison, we implement four typical methods of unsupervised clustering, including k-means clustering, the pLSA [11], [46], Affinity Propagation (AP) [19] and LDA [13]. These methods use exactly the same setting as in our approach for fairly evaluating, and the categorization number for them is manually fixed, i.e., 16 for SYSU_VIDEO_SHOTS dataset and 40 for the TRECVID dataset, respectively. Moreover, to illustrate the benefit of the proposed video brick features (i.e., EVWs and IVWs), we also compare with the HoG [21] descriptors, the state-of-the-arts appearance features for 2D image patches. In this comparison, we first extract 2D image patches from videos with a regular size, say 15×15 pixels, and then construct the word dictionary with the HoG descriptor. For all the five methods, we perform the experiments with both the video brick features and image patch features. The performance comparisons are exhibited in Fig. 7 based on the two benchmark metrics, Purity and Conditional Entropy. Overall, our method achieves the average categorization purity of 0.5647 and 0.2650 on the two datasets respectively, and outperforms the other unsupervised categorization approaches; the advantages of video brick features are clearly illustrated as well.

In our method, the clustering inference is performed simultaneously with the feature selection for category modeling. In Fig. 8, we show the selected video words of different types, i.e., structural, color, and textural words, for different categories, and the weights of top 40 informative words are plotted as well. The results are quite reasonable because the selected words indeed

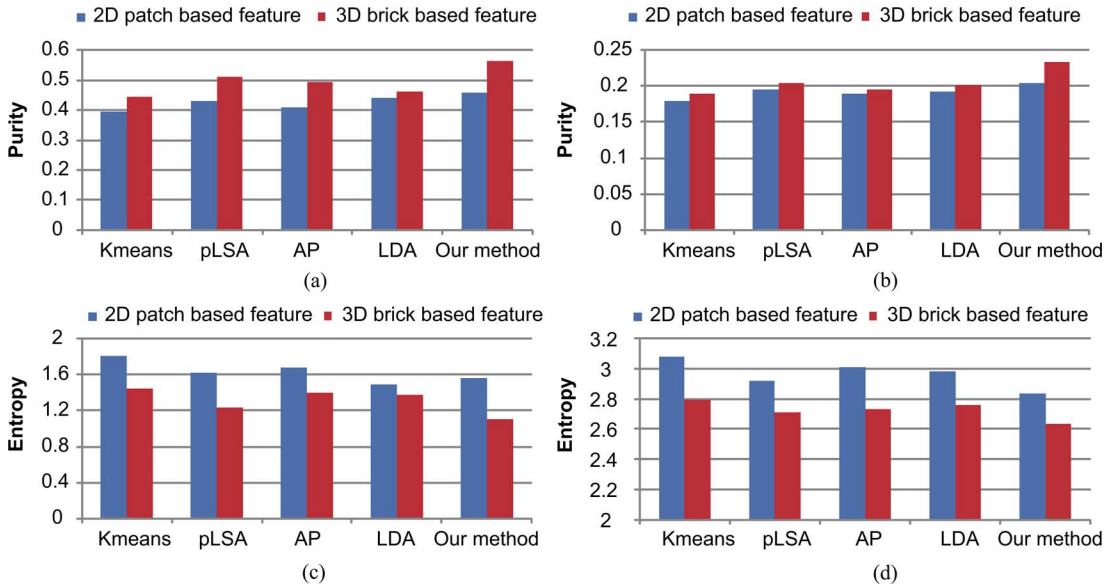

Fig. 7. Experiment results with the two benchmark metrics, Purity and Conditional Entropy. We compare our method with other popular unsupervised categorization algorithms as well as the recently proposed image features. In each figure, the vertical and horizontal axes, respectively, represent the benchmark metrics and the different clustering algorithms. The red pillars denote using the proposed video brick features and blue ones using the image patch features. (a) Purity comparison on SYSU_VIDEO_SHOTS dataset. (b) Purity comparison on TRECVID dataset. (c) Entropy comparison on SYSU_VIDEO_SHOTS dataset. (d) Entropy comparison on TRECVID dataset.

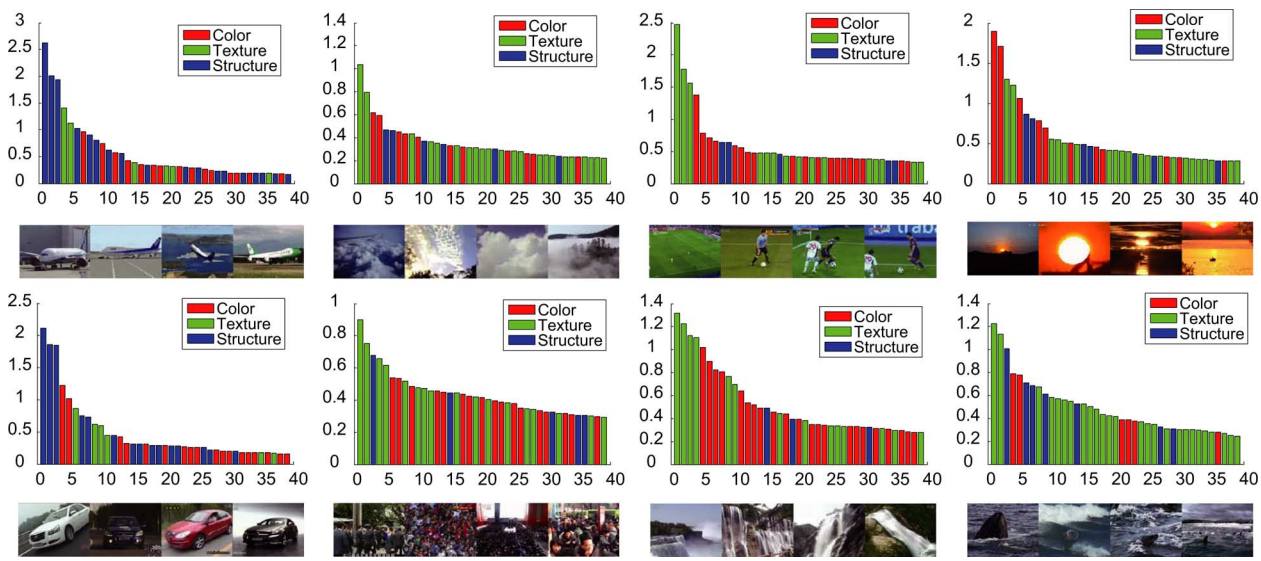

Fig. 8. The selected video words for 8 categories. For each category, we exhibit the top 40 informative videos words according to their information gain (the vertical axe). The different colors represent different types of video words.

match the appearance and motion dynamics of the video shots. For example, the structural video words mainly affect the video shots such as airplane and car; the textual video words play important roles in video shots of cloud, waterfall and the crowd; the color video words are effective to model the sunset shots.

In order to demonstrate the benefit of the compositional SWC algorithm for cluster sampling, we compare the convergence efficiency with the original 2-way SWC algorithm. Fig. 9 shows the convergence curves of the target energy, $-\log P(S|\mathbb{D})$, in the inference with the increasing iteration steps. We test on both the two datasets. The dashed (blue) curves and the solid (red) curves are generated from the original SWC algorithm and the compositional SWC algorithm, respectively.

C. Complexity Analysis

In the following, we present an elementary analysis about the complexity of our approach, along with the increase of data scale. Complexity is one of the key concerns for many multimedia processing systems and it is associated with both space and time.

In this work, the space complexity is co-related with the numbers of video words (i.e., EVWs, TVWs and CVWs), which are all fixed as constants. The EVWs are defined with a set of moving Gabor wavelets, and thus the number of EVWs is fixed as 324. For constructing the implicit video words, we randomly collect more than 40000 video bricks from the

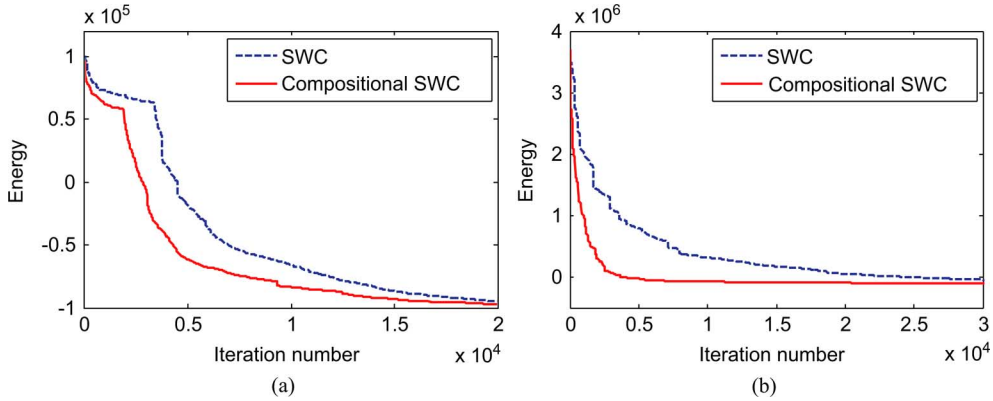

Fig. 9. Convergence comparison of the proposed compositional SWC and the original SWC algorithm. The vertical axis represents the target energy ($-\log p(S|\mathbb{D})$). Note that the energy goes inversely with the posterior probability). The horizontal axis represents the iterator steps. We observe that the compositional SWC algorithm converges faster. (a) Convergence comparison on SYSU_VIDEO_SHOTS Dataset. (b) Convergence comparison on TRECVID Dataset.

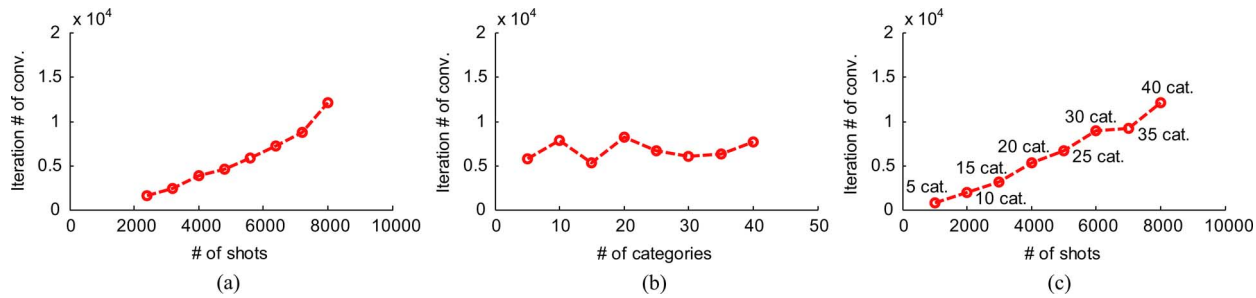

Fig. 10. Time complexity analysis with the increasing of data scale. This analysis is performed on the TRECVID dataset. In each figure, the vertical axis represents the speed (iteration number) of convergence; the horizontal axis in (a) represents the number of video shots with the fixed 40 underlying categories, in (b) the number of categories with the fixed amount of video shots, and in (c) the number video shots with various underlying categories.

databases and generate 1400 TVWs and 600 CVWs, as introduced in Section II-A. Therefore, there are totally 2324 of video words in the dictionary. This dictionary of video words is kept for carrying out all experiments. In practice, we may re-generate the video words once the data advances by an order of magnitude, i.e., hundreds of underlying categories.

We consider the time complexity as the iteration steps in the stochastic graph partition. For the experiments on the SYSU_VIDEO_SHOTS dataset, it costs average 0.09 s per iteration step, and for the TRECVID data, it costs average 0.22 s per step. In the experiments, two factors basically affect the time complexity: the total number of video shots to be categorized and the underlying category number. For quantitatively analysis, we visualize the numbers of iteration steps on various data scales, as Fig. 10 shows. We can observe that the time complexity increases in the nonexponential order, making the system is potential to be applied to larger scale data.

VII. Conclusion

This paper studies a general framework to discover video shot categories automatically via unsupervised graph partition. Compared with the previous methods, the advantages of the proposed method are identified on two public datasets and summarized as follows. First, the proposed video words are powerful to capture local appearance information and motion dynamics in the video shots. Second, the feature selection is performed simultaneously with the clustering procedure, guided by a generative model for each category. Third, we adopt a cluster sampling algorithm for efficient inference, in which the clustering number is automatically determined with the global optimization.

In the future, we plan to improve the method in two aspects. (1) More effective models, e.g., pyramid model or hierarchical model, can be utilized to represent video shots, instead of the "bag of words" model. (2) The incremental learning strategy can be integrated into pursuing category models, particularly for large-scale data.

References

[1] J. Fan, H. Luo, Y. Gao, and R. Jain, "Incorporating concept ontology for hierarchical video classification, annotation, and visualization," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 939–957, 2007.

[2] C. W. Ngo, T. C. Pong, and H. J. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 446–458, 2002.

[3] J. Fan, A. Elmagarmid, X. Zhu, W. Aref, and L. Wu, "Classview: Hierarchical video shot classification, indexing, and accessing," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 70–86, 2004.

[4] M. Szummer and R. Picard, "Indoor-outdoor image classification," in *Proc. IEEE Int. Workshop Content-Based Access of Image and Video Database*, 1998, vol. 1, pp. 42–51.

[5] A. Vailaya, M. Figueiredo, A. Jain, and H. J. Zhang, "Image classification for content-based indexing," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 117–130, Jan. 2001.

[6] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2005, vol. 2, pp. 524–531.

[7] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering object categories in image collections," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2005.

[8] Z. Xi, X. Zhuang, S. Yan, S. Chang, M. Jonhson, and T. Huang, "Sift-bag kernel for video event analysis," in *Proc. ACM Conf. Multimedia*, 2008.
[9] L. Lin, X. Liu, S. Peng, H. Chao, Y. Wang, and B. Jiang, "Object categorization with sketch representation and generalized samples," *Pattern Recognit.*, vol. 45, no. 10, pp. 3648–3660, 2012.
[10] D. Dai, T. Wu, and S. C. Zhu, "Discovering scene categories by information projection and cluster sampling," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2010.
[11] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proc. Eur. Conf. Comput. Vision (ECCV), LNCS*, 2006, vol. 3954, pp. 517–530.
[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learning Res.*, vol. 3, pp. 993–1022, 2003.
[13] C. H. Li, B. C. Kuo, and C. T. Lin, "LDA-based clustering algorithm and its application to an unsupervised feature extraction," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 152–163, 2011.
[14] P. Wang, Z. Q. Liu, and S. Q. Yang, "Investigation on unsupervised clustering algorithms for video shot categorization," *Soft Comput.—Fusion of Foundat., Methodol., Applicat.*, vol. 11, pp. 355–360, 2007.
[15] L. Y. Duan, M. Xu, Q. Tian, C. S. Xu, and J. Jin, "A unified framework for semantic shot classification in sports video," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1066–1083, 2005.
[16] A. K. Zhou, H. T. Hermans, and J. Rehg, "Movie genre classification via scene categorization," in *Proc. ACM Conf. Multimedia*, 2010.
[17] P. Gupta, S. Arrabolu, M. Brown, and S. Savarese, "Video scene categorization by 3D hierarchical histogram matching," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2009, vol. 2, pp. 1655–1662.
[18] Y. Liu and F. Wu, "Multi-modality video shot clustering with tensor representation," *Multimedia Tools Applicat.*, vol. 41, pp. 93–109, 2009.
[19] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.
[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2005.
[22] L. Lin, T. Wu, Z. Xu, and J. Porway, "A stochastic graph grammar for compositional object representation and recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1297–1307, 2009.
[23] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vision*, vol. 79, no. 3, pp. 229–318, 2008.
[24] G. Zhu, M. Yang, K. Yu, W. Xu, and Y. Gong, "Detecting video events based on action recognition in complex scenes using spatio-temporal descriptor," in *Proc. ACM Conf. Multimedia*, 2009.
[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2001.
[26] D. Liu and T. Chen, "Unsupervised image categorization and object localization using topic models and correspondences between images," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2007.
[27] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine, "Unsupervised object discovery: A comparison," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 284–302, 2010.
[28] A. Barbu and S. C. Zhu, "Generalizing Swendsen-Wang for image analysis," *J. Computat. Graphic. Statist.*, vol. 16, no. 4, pp. 877–900, 2007.
[29] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, 1984.
[30] L. Lin, X. Liu, and S. C. Zhu, "Layered graph matching with composite cluster sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1426–1442, 2010.
[31] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
[32] L. Lin, P. Luo, X. Chen, and K. Zeng, "Representing and recognizing objects with massive local image patches," *Pattern Recognit.*, vol. 45, no. 1, pp. 231–240, 2012.
[33] S. C. Zhu, K. Shi, and Z. Si, "Learning explicit and implicit visual manifolds by information projection," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 667–685, 2010.
[34] X. Liu, L. Lin, S. Yan, H. Jin, and W. Jiang, "Adaptive object tracking by learning hybrid template on-line," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1588–1599, Nov. 2011.
[35] Y. Zhao, H. Gong, and Y. Jia, "Pursuing atomic video words by information projection," in *Proc. IEEE Asian Conf. Comput. Vision (ACCV)*, 2010.
[36] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, 1997.
[37] M. Tappen and W. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, 2003.
[38] Y. Zhao, H. Gong, L. Lin, and Y. Jia, "Spatio-temporal patches for night background modeling by subspace learning," in *Proc. IEEE Int. Conf. Pattern Recognition (ICPR)*, 2009.
[39] X. Liu, L. Lin, H. Jin, and S. C. Zhu, "Trajectory parsing by cluster sampling in spatio-temporal graph," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2009.
[40] X. Zhao, Y. Fu, and Y. Liu, "Human motion tracking by temporal-spatial local Gaussian process experts," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1141–1151, Apr. 2010.
[41] E. Shechtman and M. Irani, "Space-time behavior-based correlation-or-how to tell if two underlying motion fields are similar without computing them?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 2045–2056, Nov. 2007.
[42] M. Marszalk, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, 2009.
[43] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, 1996.
[44] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Phys. Rev. Lett.*, vol. 58, no. 2, pp. 86–88, 1987.
[45] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in *Proc. 8th ACM Int. Workshop Multimedia Inform. Retrieval (MIR '06)*, New York, 2006, pp. 321–330.
[46] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, pp. 177–196, 2001.

Xiaohua Duan (M'12) received the B.B.A degree in the Department of Economic Management from Xi'an University of Posts and Telecommunications in 2004, and M.S. degree in computer science from Sun Yat-sen University in 2007. Now he is a Ph.D. candidate of Computer Science at Sun Yat-sen University, Guangzhou, China. His current research interests are image/video processing, multimedia analysis and retrieval, and computer vision.

Liang Lin (M'05) received the B.S. and Ph.D. degrees from the Beijing Institute of Technology (BIT), Beijing, China, in 1999 and 2008, respectively. From 2006 to 2007, he was a joint Ph.D. student with the Department of Statistics, University of California, Los Angeles (UCLA). He was a Post-Doctoral Research Fellow with the Center for Image and Vision Science of UCLA. From 2007 to 2009, he was a Senior Research Scientist with the Lotus Hill Research Institute, Hubei, China. He is currently an Associate Professor with the Software School of Sun Yat-Sen University, Guangzhou, China. His current research interests include but are not limited to computer vision, pattern recognition, machine learning, and multimedia technology. He has authored or co-authored over 40 academic papers over a wide range of research topics.

Dr. Lin has received a number of honors, including several scholarships while pursuing the Ph.D. degree, the Beijing Excellent Students Award in 2007, China National Excellent PhD Thesis Award Honorable Mention in 2010, and the Best Paper Runners-Up Award in ACM NPAR 2010.

Hongyang Chao (M'06) received the B.Sci. degree from Sun Yet-sen University, Guangzhou, China, and the Ph.D. degree from Sun Yet-sen University, Guangzhou, China, both in Computational Mathematics.

From 1988 to 1990, she joined the Department of Computer Science of the Sun Yet-sen University at Guangzhou, China, where she was an Assistant Professor. In November 1990, she became an Associate Professor of Computer Science and the director of the Institute of Computational Mathematics. From 1994–1995, she visited the Department of Computer Science at Stanford University as a research scholar under the support of the Lingnan Foundation in the US. Subsequently, she visited the Department of Computer Science of the University of North Texas as a visiting professor from June 1995 to April 1998. During this period, she joined a software company named Infinop (later acquired by Vianet/ESPRE), where she was also a founding researcher. She continuously worked in the same company as a Chief Scientist until 2004. Afterward, she joined the School of Software at Sun Yet-sen University in Guangzhou, China, where she was an Administrative Deputy Dean for the school from 2004 to 2008. She was awarded by the "Hundred Talents Program" of the University in 2004. She presently is an Associate Dean and a full Professor in the school. She has published extensively in the area of image/video processing and holds three patents. Her current research interests include the areas of image and video processing, image and video compression, massive multimedia data analysis and understanding, and content based image (video) retrieval.