# Adaptive Scene Category Discovery With Generative Learning and Compositional Sampling

Liang Lin, *Member, IEEE*, Ruimao Zhang, and Xiaohua Duan

***Abstract*— This paper investigates a general framework to discover categories of unlabeled scene images according to their appearances (i.e., textures and structures). We jointly solve the two coupled tasks in an unsupervised manner: 1) classifying images without predetermining the number of categories and 2) pursuing generative model for each category. In our method, each image is represented by two types of image descriptors that are effective to capture image appearances from different aspects. By treating each image as a graph vertex, we build up a graph and pose the image categorization as a graph partition process. Specifically, a partitioned subgraph can be regarded as a category of scenes and we define the probabilistic model of graph partition by accumulating the generative models of all separated categories. For efficient inference with the graph, we employ a stochastic cluster sampling algorithm, which is designed based on the Metropolis–Hasting mechanism. During the iterations of inference, the model of each category is analytically updated by a generative learning algorithm. In the experiments, our approach is validated on several challenging databases, and it outperforms other popular state-of-the-art methods. The implementation details and empirical analysis are presented as well.**

***Index Terms*— Generative learning, graph partition, scene understanding, unsupervised categorization.**

## I. Introduction

CATEGORY discovery for unlabeled images is an important research topic with a wide range of applications, such as content-based image retrieval [1], [2], image database management [3], [4], and scene understanding [5]–[7]. In this paper, we develop a unified framework to categorize scene images in an unsupervised manner. Specifically, with this framework, a batch of unlabeled scene images can be automatically grouped into different categories according to their contents, and we simultaneously generate the probability models for the categories.

We pose the unsupervised image categorization as a graph partition task, i.e., each generated partition indicates a potential category; then, we employ a novel clustering sampling algorithm for inference, which is an extension of Swendsen–Wang cuts (SWCs) [32] for greatly improving the inference efficiency. More specifically, the graph partition is formulated under a probabilistic framework that accumulates the generative models of all categories. Intuitively, the goodness of partitions is determined based on how well the learned models explain or generate the partitioned categories. Therefore, solving the optimal graph partition is equivalent to searching the maximum probability.

Natural scenes usually contain diverse image contents related with different types of visual appearance patterns, e.g., inhomogeneous (or structural) textures (buildings, cars, roads, etc.) and homogeneous textures (grasses, water surfaces, etc.) [19]. Many studies [20]–[22] on designing image features show that the distribution-based descriptors [e.g., SIFT [23], histogram of oriented gradients (HOG) [24], and Textons [11]] and the binary operators [e.g., local binary pattern (LBP) and its variants [25], [26]] lead to state of arts on representing low-level image contents from different aspects. The former features tend to well describe the inhomogeneous textures, while the latter can be applied to capture highly random textures [27], [38]. Therefore, in our method, we represent an image with a number of image patches at multiple scales. Two effective image features, the HOG [24] and the center-symmetric LBP (CS-LBP) [26], are employed to describe the image patches. Specifically, we define two types of visual words [i.e., inhomogeneous textural words (ITWs) and homogeneous textural words (HTWs)], respectively, based on the two features. In the literature, the significance of using combined features is also demonstrated in various vision tasks, e.g., near-duplicate image retrieval [1], [2], object detection [35], and video tracking [36], [37].

Moreover, we adaptively select informative features (i.e., visual words) for each scene class, along with the categorization procedure. Several methods of image categorization [9], [10] show that different categories of images are probably captured by different class-specific features. Some discriminative learning algorithms (e.g., Adaboost [28] and Support Vector Machines) perform very well in feature selection. However, they are not suitable for our task, since these algorithms rely on negative data and are often sensitive to outliers. In contrast, our framework employs a generative learning algorithm based on

Manuscript received October 8, 2013; revised January 11, 2014, February 18, 2014, and March 2, 2014; accepted March 8, 2014. Date of publication March 26, 2014; date of current version February 4, 2015. This work was supported in part by the Hi-Tech Research and Development (863) Program of China under Grant 2013AA013801, in part by the Guangdong Science and Technology Program under Grant 2012B031500006, in part by the Guangdong Natural Science Foundation under Grant S2013050014548, in part by the Special Project on Integration of Industry, Education and Research of Guangdong Province under Grant 2012B091100148, and in part by the Fundamental Research Funds for the Central Universities under Grant 13lgjc26. This paper was recommended by Associate Editor S. Battiato.

L. Lin is with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Sun Yat-Sen University, Guangzhou 510006, China; the School of Advanced Computing, Sun Yat-Sen University, Guangzhou 510006, China; and also with the SYSU-CMU Shunde International Joint Research Institute, Guangzhou 510006, China (e-mail: linliang@ieee.org).

R. Zhang and X. Duan are with Sun Yat-Sen University, Guangzhou 510006, China.

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/TCSVT.2014.2313897

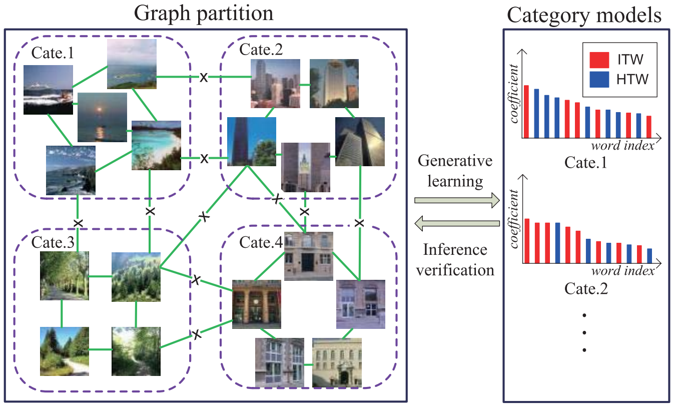

Fig. 1. Overview of our framework. We formulate the problem of image category discovery as a graph partition task. In the left panel, the images are treated as graph vertices that are partitioned into subgraphs by turning OFF the graph edges. As shown in the right panel, the generative models for all partitioned categories are pursued simultaneously, and the models are also used to guide the inference of graph partition. The models are learned with two types of visual words: ITWs and HTWs defined based on two image descriptors.

the MaxMin pursuit framework [29], so that we can fast pursue the generative models of categories without extra negative data.

The framework of our approach is shown in Fig. 1. The key contribution of this paper is a general approach for automatic scene image categorization, in which the cluster (i.e., category) number is automatically determined. The generative category models are learned and updated simultaneously together with the categorization procedure. Our method is evaluated on several public data sets and outperforms the state-of-the-art approaches. It is worth mentioning that the graph partition and category models are 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.

### A. Related Work

Most of the methods of scene image categorization involve a procedure of supervised learning, i.e., training a multiclass predictor (classifier) with the manually labeled images [8]. Unsupervised image categorization is often posed as clustering images into groups according to their contents (i.e., appearances and/or structures). In some traditional methods [9], various low-level features (such as color, filter banks, and textons [11], [30]) are first extracted from images, and a clustering algorithm (e.g., $k$-means or spectral clustering) is then applied to discover categories of the samples.

To handle diverse image content, some effective image representations, such as bag of words (BoWs), are proposed [12], [13], and they represent an image using a pretrained collection (i.e., dictionary) of visual words. Furthermore, Lazebnik *et al.* [14] present a spatial pyramid representation of BoWs by pooling words at different image scales, and this representation effectively improves the results for scene categorization [15]. Farinella and Battiato [16] propose to build an effective scene representation based on constrained and compressed domains.

To exploit the latent semantic information of scene categories, Bosch *et al.* [17] discuss the probabilistic latent semantic analysis (pLSA) model that can explain the distribution of features in the image as a mixture of a few semantic topics. As an alternative model for capturing latent semantics, the latent Dirichlet allocation (LDA) model [18] was widely used as well.

On the other hand, the category number is required to be predetermined or be exhaustively selected in many previous unsupervised categorization approaches [7], [31]. In computer vision, the stochastic sampling algorithms [32], [33], [37] are shown to be capable of flexibly generating new clusters, merging, and removing the existing clusters in a graph representation. Motivated by these works, we propose to automatically determine the number of image categories with the stochastic sampling.

The rest of this paper is organized as follows. We first introduce the image representation in Section II. Then, we present the problem formulation in Section III, and follow with a description of the inference algorithm for unsupervised image categorization in Section IV. Section V discusses the learning algorithm for category model pursuit during the inference procedure. The experimental results and comparisons are exhibited in Section VI, and this paper is concluded in Section VII.

## II. Image Representation

In this section, we start by briefly introducing the two effective low-level image descriptors used in this paper, and define two types of visual words to construct the dictionary of images.

Previous works on designing image features can be roughly divided into two categories [27], [35]. The first one explicitly describes images with local gradients that are sensitive to structures (e.g., edges, boundaries, and junctions) and distinct textures (e.g., regions of clear details). The other one reflects uncertain differences among pixels, and thus tends to be suitable for incognizable random textures (e.g., complex regions, and cluttered patterns). Thus, we utilize two typical image descriptors, i.e., HOG [24] and CS-LBP [26], respectively, in this paper. Following the studies on image representation [27], we refer a visual word $\omega$ as an ensemble or equivalence class of image patches that share the similar appearances. Letting $h(\cdot)$ be the histogram of an image feature, we define $\omega$ as

$$\omega = \left\{\Lambda : h(\Lambda) = \hat{h} + \epsilon\right\} \qquad (1)$$

where $\hat{h}$ denotes the mean histogram of the image patches and $\epsilon$ is the statistical fluctuation, i.e., a very small value. According to the two image descriptors, we define two types of visual words, ITWs and HTWs, together with the two descriptors. The benefit of combining the two types of words will be demonstrated in the experiments.

To define ITWs, the input image domain is divided into a number of regular cells; at each pixel, a local gradient is calculated, and a histogram is pooled over each cell for different orientations. As shown in Fig. 2, we decompose an image patch by 2 × 2 cells and quantize the orientations into eight angles. The dimension of this descriptor is thus 32.

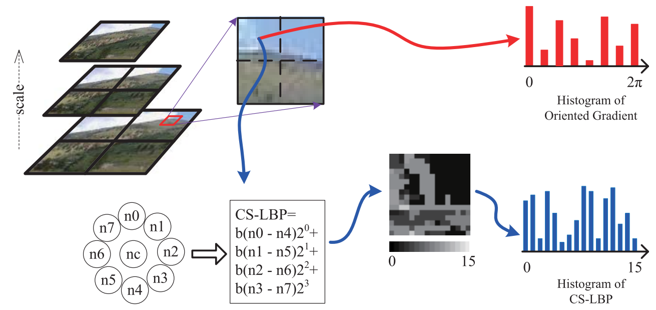

Fig. 2. Image representation. We represent an image with the pyramid BoW model with two types of visual words that are, respectively, defined based on two image descriptors, i.e., HOG [24] and CS-LBP [26].

The HTWs are generated using the CS-LBP operator, which is computed at every pixel in the input image domain. It compares center-symmetric pairs of the given pixel and forms a binary vector. Given a pixel located at $x$ with $\hat{n} = 8$ neighborhood pixels that are equally spaced on a circle of radius, as the example shown in Fig. 2, the binary vector can be calculated as

$$\sum_{i=0}^{\hat{n}/2-1} b(n_i - n_{i+\hat{n}/2})2^i, \qquad b(x) = \begin{cases} 1, & x > 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $n_i$ and $n_{i+\hat{n}/2}$ correspond to the intensity scales of center-symmetric pairs of pixels. We compute the operator over all pixels in the domain; the obtained binary vectors can be converted into decimal strengths in the range of [0, 15]. An example of a strength map is shown in Fig. 2. Since there are four cells divided, we further pool the strengths into a histogram with $16 \times 4 = 64$ bins, denoted as $h^b$.

Then, we construct the dictionary to represent images with the visual words. In our implementation, we collect a large number of image patches from our database and compute the two descriptors for each, and group them into a batch of clusters (words) using the $k$-means algorithm. Thus, we obtain a dictionary $\mathcal{W} = \{\omega_i, i = 1, \ldots, m\}$, where $\omega_i$ is a visual word (i.e., ITW or HTW).

Given an image $\mathbf{I}$, we represent it with a spatial pyramid format, $1 + 4 \times 2 = 9$ blocks, i.e., three scales (resolutions) and four blocks in each scale except the top, as shown in Fig. 2. In each block, the image domain is further decomposed into regular image patches that are mapped to the generated words. The image of a block $\mathbf{J}$ can be thus represented as a vector using the dictionary, $(r_1(\mathbf{J}), r_2(\mathbf{J}), \ldots, r_m(\mathbf{J}))$, where $r_i(\mathbf{J})$ is the response with the visual word $\omega_i$, and

$$r_i(\mathbf{J}) = \psi\left(\sum_{\Lambda \in \mathbf{J}} \mathbf{1}_{\omega_i}(\Lambda)\right) \tag{3}$$

where $\mathbf{1}_{\omega_i}(\Lambda) = \{1|0\}$, the indicator function, is used to indicate whether the image patch $\Lambda \in \mathbf{J}$ matches with $\omega_i$. The matching is measured by either of the two descriptors, $h^a$ and $h^b$, according to the type of word $\omega_i$. Thus, we use $\sum_{\Lambda \in \mathbf{J}} \mathbf{1}_{\omega_i}(\Lambda)$ to indicate the number of the visual word $\omega_i$ matching with the image block $\mathbf{J}$. Here, $\psi(\cdot)$ is the sigmoid function $\delta(\cdot)$ that is characterized by a saturation level.

The image $\mathbf{I}$ is hence represented as $\mathcal{R}(\mathbf{I})$, by concatenating the vectors of all nine blocks.

## III. Problem Formulation

Given a set of unlabeled images $\mathcal{D}$, the goal of our framework is to categorize them into an unknown number of disjoint $K$ clusters as

$$\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\} \tag{4}$$

where $\cup_{k=1}^{K} \pi_k = \mathcal{D}$, $\pi_i \cap \pi_j = \emptyset$, $\forall i \neq j$.

We first build a graph $G_0 = \langle V, E_0 \rangle$, in which $V = \mathcal{D} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$ is the set of graph vertices specifying the images to be categorized, and $E_0$ is the set of edges connecting neighboring graph vertices. Then, we solve the task of graph partition by cutting edges of the graph, i.e., generating disjoint subgraphs. However, $G_0$ is a fully connected graph where the initial edge set $E_0$ could be very large. To reduce computational complexity, we shall compute a relatively sparse graph representation $G_0 = \langle V, E \rangle$ by pruning edges, $E \subset E_0$.

For any edge $e \in E_0$, an auxiliary connecting variable $\mu_e = \{\text{ON}|\text{OFF}\}$ is first introduced, which indicates whether the edge is turned ON or OFF. Then, we can define the edge connecting probability by measuring the similarity of two connected graph vertices. In our implementation, We define the similarity using the visual words $\mathcal{W}$. Specifically, for any vertices $v \in V$, we represent it as $\mathcal{R}(\mathbf{I}) = (r_1(\mathbf{I}), r_2(\mathbf{I}), \ldots, r_m(\mathbf{I}))$, where $r_i(\mathbf{I})$ is the response of the word $\omega_i$, as in (3). Thus, we can define the connecting probability $q_e$ for two arbitrary images $\mathbf{I}_s \in V, \mathbf{I}_t \in V$ as

$$q_e(s,t) = p(\mu_e = on|v_s, v_t) = \exp\left\{ -\tau\left[\mathcal{KL}(\mathcal{R}_s \| \mathcal{R}_t)\right]\right\} \tag{5}$$

where we denote $\mathcal{R}_s = \mathcal{R}(\mathbf{I}_s)$ and $\mathcal{R}_t = \mathcal{R}(\mathbf{I}_t)$ for notation simplicity. $\mathcal{KL}()$ is the symmetric Kullback–Leibler distance for measuring two feature vectors. $\tau$ is a constant parameter. $q_e(s,t)$ should be close to zero if $\mathbf{I}_s$ and $\mathbf{I}_t$ naturally belong to different categories; the edge $e$ connecting $\mathbf{I}_s$ and $\mathbf{I}_t$ could be then turned OFF with high probability.

In practice, the edges with very low turn-ON probability can be directly removed. Furthermore, we enforce that each vertex can be only connected to at most six neighbors. That is, for any vertex, we keep six edges with the highest connecting probabilities, and remove the other edges. Therefore, we obtain the sparse graph $G = \langle V, E \rangle$, where $E \subset E_0$.

With the graph representation, we pursue the generative probability models for all categories as

$$\Phi = \left\{\phi_k(\mathbf{I}; W_k, \Theta_k), W_k \subset \mathcal{W},\ k = 1, \ldots, K\right\} \tag{6}$$

where $W_k \subset \mathcal{W}$ denotes the selected visual words for modeling the category $\pi_k$ and $\Theta_k$ includes the corresponding model parameters, i.e., the coefficients of words. The overall solution of image category discovery can be defined as

$$S = (K, \Pi, \Phi) \tag{7}$$

where $K$ is the inferred category number. The graph partition $\Pi$ and category modeling $\Phi$ can be solved together in a Bayesian inference framework. Assume that $p(S)$ and $p(\mathcal{D}|S)$ denote the prior model and the likelihood model, respectively.

$p(S)$ can be simply modeled by incorporating an exponential function for $K$, as we impose no priors on $\Pi$ and $\Phi$. The likelihood model $p(\mathcal{D}|S) = p(\mathcal{D}|\Pi, \Phi)$ can be defined as a product of generative models of all separated categories, as we assume that the models are generated independently to each other. We can then define the posterior probability of solution $S$ as

$$\begin{aligned} p(S|\mathcal{D}) &\propto p(S)p(\mathcal{D}|S) \\ &= \exp\{-\beta K\} \prod_{k=1}^{K} \phi_k(\mathbf{I}; W_k, \Theta_k) \end{aligned} \tag{8}$$

where $\beta$ is an empirical parameter for constraining the number of inferred categories. The category model $\phi_k(W_k, \Theta_k)$ is defined on the probabilistic distribution of the images in partition $\pi_k$. The models for all categories can be learned and updated during the procedure of image categorization.

## IV. Inference for Image Categorization

The objective of inference is to search for the optimized solution $S^*$ by maximizing the posterior probability in (8)

$$S^* = \arg\max p(S|\mathcal{D}). \tag{9}$$

This optimization is very challenging due to two characters in our problem: 1) the unknown number of partitions and 2) no confident initializations, i.e., lack of the initial category models. Therefore, we employ the stochastic sampling algorithm instead of using deterministic inference algorithms.

In the research area of stochastic inference, cluster sampling is very powerful for simulating Ising/Potts graphical models, which is designed under the Metropolis–Hasting mechanism. Recently, Barbu and Zhu [32] generalized the algorithm, namely SWCs, to solve graph partition in several vision applications. This algorithm enables us to effectively search for the maximum of posterior probability. It simulates a Markov chain containing a sequence of states in the solution space $\Omega$ and visits the Markov chain by realizing a reversible jump between any two successive states.

In the following, we first introduce the SWC algorithm, and then discuss an extension [34] that greatly improves the inference efficiency. In general, the SWC algorithm iterates in two steps.

1) Generate the connected components (CPs) by probabilistically turning OFF the connecting edges in the graph. Graph vertices connected together by the on edges form a CP. Specifically, any two vertices in one CP are linked by a path that consists of several edges. For arbitrary edge $e \in E$, we sample its connecting variable $\mu_e$ and decide it is turned ON or OFF in this step. Then, we obtain a few CPs, each of which is a set of connected graph vertices.
2) Explore a new partition solution by relabeling one of the CPs. Assume that the current partition solution is $S_A$ and we are exploring a new solution $S_B$. Given one randomly selected CP, the reversible operators are developed to reassign its label. For example, the selected CP can be merged into current separated category by receiving the same label with the category; otherwise, a new category can be created if the selected CP receives a new label.

We design the algorithm by the Metropolis–Hastings mechanism [32]. 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 moving from $S_A$ to $S_B$ is

$$\alpha(S_A \rightarrow S_B) = \min\left(1, \frac{Q(S_B \rightarrow S_A)}{Q(S_A \rightarrow S_B)} \cdot \frac{p(S_B|\mathcal{D})}{p(S_A|\mathcal{D})}\right). \tag{10}$$

For any state transition, the proposal probability usually involves two aspects: 1) the generation of CP and 2) the label assignment of CP. In our method, we make the CP be assigned randomly with a uniform distribution, so that the proposal probability can be simplified. Thus, the ratio of proposal probability is calculated by

$$\frac{Q(S_B \rightarrow S_A)}{Q(S_A \rightarrow S_B)} = \frac{\prod_{e\in C_B}(1-q_e)}{\prod_{e\in C_A}(1-q_e)} \tag{11}$$

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 $S_B$. Here, we name $C_A$ or $C_B$ as a cut, following [32].

To further accelerate the convergence of inference, we employ an improved version of the SWC algorithm that was originally proposed by us for video shot categorization [34]. In the original algorithm, only one CP is selected and processed in each step of solution exploration. In our method, we process a number of CPs together by coupling them into a combinatorial cluster. We thus regard this algorithm as the compositional SWC (CSWC). The CSWC algorithm is able to enlarge the searching scope during the sampling iterations, resulting in faster convergence than the original version.

Fig. 3 shows the idea of CSWC. Given a current state $S_A$ [Fig. 3(a)], we can generate a number of CPs by turning OFF a few edges [Fig. 3(b)]. Then, we construct a higher layer graph $\mathbf{G}$ based on these CPs. In this graph, we treat each CP as a vertex, and link any two neighboring CPs by an edge, as shown in Fig. 3(c). Within $\mathbf{G}$, we can generate and select several combinatorial clusters to explore the new solution.

Similar with the definitions in $G$, we calculate the turn-ON probability $q^{\mathrm{CP}}$ for an edge in $\mathbf{G}$ according to the similarity of two connected vertices (i.e., CPs), which can be derived from the original graph $G$. Specifically, given two neighboring CP$_i$ and CP$_j$, we measure their similarity by aggregating all the edges in $G$ that connects the vertices in $G$ belonging to CP$_i$ and CP$_j$, respectively. Thus, we define the connecting probability of an edge in $\mathbf{G}$ as

$$\begin{aligned} q^{\mathbf{G}} &\propto \left[1 - \prod(1-q_e)\right] \\ e &= \langle s, t\rangle, \quad s \in CP_i, \quad t \in CP_j. \end{aligned} \tag{12}$$

By probabilistically turning OFF the edges in $\mathbf{G}$, we can also generate several CPs, and we regard them as combinatorial clusters to distinguish the CPs in $G$. In Fig. 3(d), four combinatorial clusters are generated. Different with the algorithm in [34], we allow more than one combinatorial clusters to be selected in this step, and we assign labels to the them.

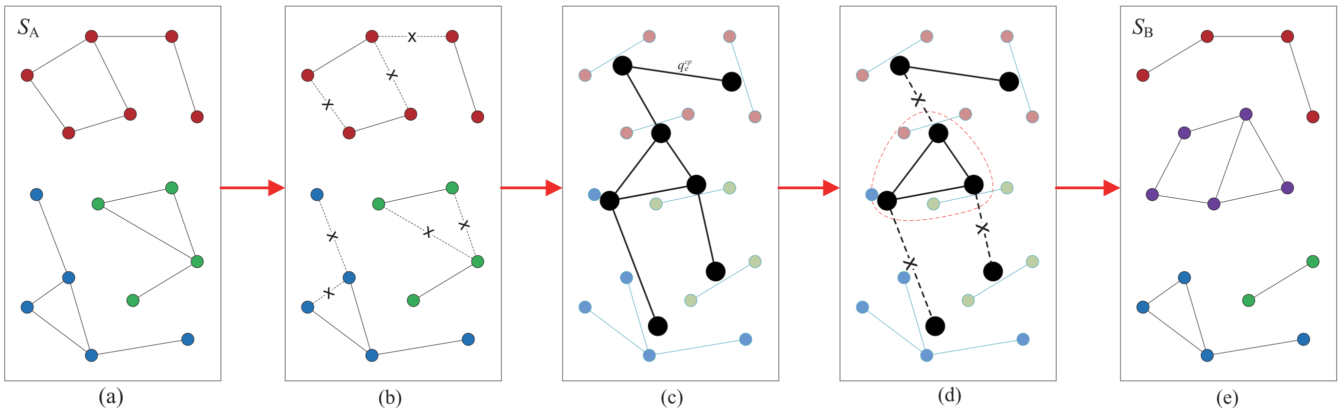

Fig. 3. Illustration of the CSWC algorithm for exploring a new solution state. (a) Current solution state $S_A$. (b) Sample ON/OFF edge in each CP. (c) Build a higher layer graph. (d) Sample ON/OFF edges on the higher layer graph. (e) New solution state $S_B$.

In this way, we generate a new solution of graph partition accordingly. In the implementation, we enforce each combinatorial cluster being processed as a atomic unit, i.e., all original CPs in the compositional cluster will receive the same label. As Fig. 3 shows, to go from $S_A$ to $S_B$, the original SWC algorithm needs at least three steps, whereas for CSWC, there is only one step. Note that we visualize only one selected CP in Fig. 3(d) for illustration.

During the inference, the posterior probability $p(S|\mathcal{D})$ can be changed, as we keep the category models updated with the categorization operation. Note that we only need to update the models of the categories where we add or remove images within them. We will introduce the category model learning in the following section.

## V. Category Model Learning

Given a fixed graph partition $\Pi$, we learn the probability model $\phi_k(W_k, \Theta_k)$ for each category by selecting the most informative visual words. Since all scene images in $\mathcal{D}$ are unlabeled, and no extra negative samples are provided, we employ an efficient generative learning algorithm for this task, namely MaxMin pursuit [29]. Similar approaches of combining generative learning in unsupervised categorization are discussed in [10].

Suppose the category $\pi_k$ is governed by an underlying target model $\phi_{f,k}$, the model pursuit can be solved by additively searching for a sequence of features, starting from an initial model $\phi_{k,0}$. At each step $t$, the model $\phi_{k,t}$ is updated to gradually approach $\phi_{f,k}$. Here, we ignore $k$ for notation simplicity. In the manner of stepwise pursuit, the new model $\phi_t$ is updated by adding a new feature $\omega_t$ based on the current model $\phi_{t-1}$, and $\omega_t$ imposes an additive constraint as

$$\phi_t = \frac{1}{z_t}\phi_{t-1}e^{\lambda_t r_t}$$
$$\text{s.t.}\quad E_{\phi_t}[r_t] = E_{\phi_f}[r_t] \tag{13}$$

where $r_t$ denotes the response of the word $\omega_t$. $E_{\phi_f}(r_t)$ represents the expectation of feature $\omega_t$ over the underlying model, which can be calculated by averaging feature responses over positive samples. $E_{\phi_t}[r_t]$ denotes the feature expectation on the new model. Following [29], we can derive the probability model by $T$ rounds of model pursuit as the following Gibbs form:

$$\phi(\mathbf{I}; W, \Theta) = \phi_0(\mathbf{I})\frac{1}{Z}\exp\left\{\sum_{t=1}^{T}\lambda_t r_t(\mathbf{I})\right\} \tag{14}$$

where $Z = \prod z_t$ and $\Theta = (\lambda_1, \ldots, \lambda_T)$. $z_t$ normalizes the sum of the probability to one, and $\lambda_t$ is the coefficient weight of the selected feature $\omega_t$. In our implementation, we specify the initial model $\phi_0$ as a uniform distribution over all words.

With this definition in (14), the model is updated by solving $\lambda_t$ and $r_t$ at each round $t$. Here, we discuss the MaxMin-KL algorithm for this goal, which iteratively performs with two following steps.

*Step 1 Max-KL:* The most informative feature $r_t^*$ is selected to update the current model. This step optimizes the following problem, given the candidate features

$$\begin{aligned} r_t^* &= \arg\max_{r_t} \mathcal{K}(\phi_t \| \phi_{t-1}) \\ &= \arg\max_{r_t} \lambda_t E_{\phi_f}[r_t] - \log z_t. \end{aligned} \tag{15}$$

This step could be computational expensive as we need to sample the model distribution $\phi_{t-1}$ of the previous round $t-1$. Following recent works on image template learning, we can simplify the computation by enforcing the visual words have a little overlap. In particular, all features can be selected independently. The optimization in (15) can be approximated as

$$r_t^* = \arg\max_{r_t} E_{\phi_f}[r_t] - E_{\phi_0}[r_t] \tag{16}$$

where $E_{\phi_0}[r_t]$ can be ignored, as it is a constant calculated on the initial model $\phi_0$. We calculate $E_{\phi_f}[r_t]$ by the mean response values

$$E_{\phi_f}[r_t] = \frac{1}{n_k}\sum_{i=1}^{n_k} r_t(\mathbf{I}_i) \tag{17}$$

where $n_k$ is the number of images belonging to the $k$th category.

**Algorithm 1**: Sketch of Our Approach

```
Input: Image data set $\mathcal{D} = \{\mathbf{I}_1, \dots, \mathbf{I}_N\}$, and visual
        words $\mathcal{W} = \{\omega_1, \dots, \omega_M\}$
Output: The categorization solution $S = \big(K, \Pi, \Phi\big)$
1. Initialization;
    (1) Represent each image $\mathbf{I}_i$ with the visual words,
    $\mathcal{R}(\mathbf{I}_i) = \{r_1(\mathbf{I}_i), \dots, r_m(\mathbf{I}_i)\}$.
    (2) Create the graph $G_0 = \langle V, E_0 \rangle$, and compute the
    turn-ON probability $q_e$ according to (5), $\forall e \in E_0$.
    (3) Remove the edges with low turn-ON probability
    deterministically, and generate the sparse graph
    $G = \langle V, E \rangle$.
2. Repeat for cluster sampling;
    (1) At the current solution $S_A$, generate the $CPs$ by
    probabilistically turning OFF connecting edges in the
    graph $G$.
    (2) Construct a high layer of graph $\mathbf{G}$ based on CPs.
    (3) Generate combinatorial clusters by
    probabilistically turning OFF edges in $\mathbf{G}$.
    (4) Select several combinatorial clusters and reassign
    labels to them.
    (5) Accept the new solution $S_B$ according to the
    acceptance rate defined in (10).
    (6) Update the generative models, $\phi(\mathbf{I}_{k,i}; W_k, \Theta_k)$, for
    the categories that have been modified according to
    solution $S_B$.
    (7) Update the posterior probability $(S|\mathcal{D})$ accordingly.
3. Output the final solution $S^* = \arg\max p(S|\mathcal{D})$.
```

*Step 2 Min-KL:* Given the selected feature $r_t$, this step is to compute its corresponding weight $\lambda_t$ and normalization term $z_t$ by

$$\begin{aligned} \lambda_t^* &= \arg\min_{\lambda_t} \mathcal{K}(\phi_t \| \phi_{t-1}) \\ \text{s.t.} \quad E_{\phi_t}[r_t] &= E_{\phi_f}[r_t]. \end{aligned} \tag{18}$$

This optimization in (18) can be solved analytically according to the proof in [27], and we conduct that

$$\begin{aligned} \lambda_t &= \log \frac{E_{\phi_f}[r_t](1 - E_{\phi_0}[r_t])}{(1 - E_{\phi_f}[r_t])E_{\phi_0}[r_t]} \\ z_t &= \exp \lambda_t E_{\phi_0}[r_t] + 1 - E_{\phi_0}[r_t]. \end{aligned} \tag{19}$$

Since we can analytically pursue this model by selecting a number $T$ of informative features, the model in (14) can be further simplified into the following form:

$$\phi(\mathbf{I}; \Theta) = \phi_0(\mathbf{I}) \prod_t^T \left[ \frac{1}{z_t} \exp\{\lambda_t r_t(\mathbf{I})\} \right]. \tag{20}$$

The proposed algorithm in the above is simple and fast, because the value of $E_{\phi_f}[r_t]$ and $E_{\phi_0}[r_t]$ for each feature only needs to be computed once in the offline stage. Hence, we can embed the learning algorithm to keep the category model updated during the iterating procedure of categorization.

Algorithm 1 summarizes the overall sketch of our framework.

TABLE I

INFERRED CLUSTER NUMBER IN EACH TIME OF EXPERIMENT

| # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| I | 8 | 9 | 8 | 8 | 9 | 10 | 10 | 11 | 8 | 9 |
| II | 9 | 10 | 10 | 11 | 10 | 9 | 12 | 11 | 12 | 10 |
| III | 16 | 17 | 15 | 16 | 16 | 15 | 18 | 16 | 15 | 17 |
| IV | 27 | 24 | 24 | 26 | 24 | 25 | 24 | 26 | 25 | 25 |

#: No. of experiments;
I: Experiments on the MIT database;
II: Experiments on the Corel database;
III: Experiments on the UIUC database;
IV: Experiments on the mixed dataset.

## VI. EXPERIMENTS

In the experiments, we apply our method to discover categories for a batch of unlabeled images with diverse appearances, and compare with other state-of-the-art approaches.

### A. Data Sets and Metrics

We use three challenging public databases for validation: 1) MIT-Scene[1]; 2) Corel[2]; and 3) UIUC-Scene[3]. Moreover, these three databases are mixed together as a larger testing set for further evaluation.

The MIT-Scene database contains 2688 images classified into eight categories according to their meaningful semantics: coasts, forest, mountains, country, highways, city views, buildings, and streets. The number of images in each category is in the range of 260–410, and the resolution of each image is 256 × 256 pixels. The Corel data set includes 1000 natural scenes with the resolution 256 × 384 pixels of 10 semantic categories: bus, coasts, dinosaurs, elephants, flower, food, horses, mountains, people, and temples. Each category contains 100 images. The UIUC-Scene database, which is an extension of MIT-Scene, contains 4485 images classified into 15 categories, and their themes are various, e.g., mountains, forest, offices, and living rooms. The mixed data set is the union of all the three databases, including totally 5485 images of 23 categories. Note that there are a few overlapping categories among them.

The usual evaluation metric for categorization is average precision, and the number of categories is assumed to be predetermined. In this paper, we adopt the two recently proposed metrics for unsupervised categorization [7], [34], i.e., purity and conditional entropy. In brief, the larger value of purity implies the better performance in categorization and conditional entropy inversely.

For the input set $\mathcal{D}$, including a number of $N$ images, suppose that the underlying category number is $L$ and the corresponding groundtruth category labels are denoted by $X = \{x_i \in [1, L], i = 1, \dots, N\}$. A testing system groups the images into $K$ categories, $\{D_k, k = 1, \dots, K\}$, with the inferred category labels $Y = \{y_i \in [1, K], i = 1, \dots, N\}$. It is worth mentioning that $K$ could be not equal to $L$, as we allow the algorithm to automatically determine the number

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/

[2] http://wang.ist.psu.edu/docs/related.shtml

[3] http://www-cvr.ai.uiuc.edu/ponce_grp/data/index.html

TABLE II

PERFORMANCE COMPARISON VIA PURITY (HIGHER IS BETTER)

| | K-means | GIST | pLSA | LDA | AP | Ours | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ITW+HTW | ITW | HTW |
| MIT | 0.5529 | 0.5770 | 0.6457 | 0.6096 | 0.5546 | **0.6721** | 0.5764 | 0.6000 |
| Corel | 0.5337 | 0.5644 | 0.6070 | 0.5980 | 0.5612 | **0.6203** | 0.6160 | 0.6040 |
| UIUC | 0.4487 | 0.4514 | 0.5074 | 0.5449 | 0.5850 | **0.5964** | 0.5613 | 0.5148 |
| Mixed | 0.3632 | 0.3801 | 0.4136 | 0.4801 | 0.5017 | **0.5295** | 0.4836 | 0.4226 |

TABLE III

PERFORMANCE COMPARISON VIA CONDITIONAL ENTROPY (LOWER IS BETTER)

| | K-means | GIST | pLSA | LDA | AP | Ours | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ITW+HTW | ITW | HTW |
| MIT | 1.2465 | 1.2102 | 1.0156 | 1.1836 | 1.1400 | **0.8963** | 1.1536 | 1.1145 |
| Corel | 1.3105 | 1.2136 | 1.1234 | 1.1371 | 1.2577 | **1.0909** | 1.1036 | 1.1154 |
| UIUC | 1.5020 | 1.4564 | 1.4322 | 1.3146 | 1.2121 | **1.1581** | 1.2150 | 1.3948 |
| Mixed | 1.7603 | 1.7172 | 1.6811 | 1.5828 | 1.5127 | **1.4328** | 1.4971 | 1.5955 |

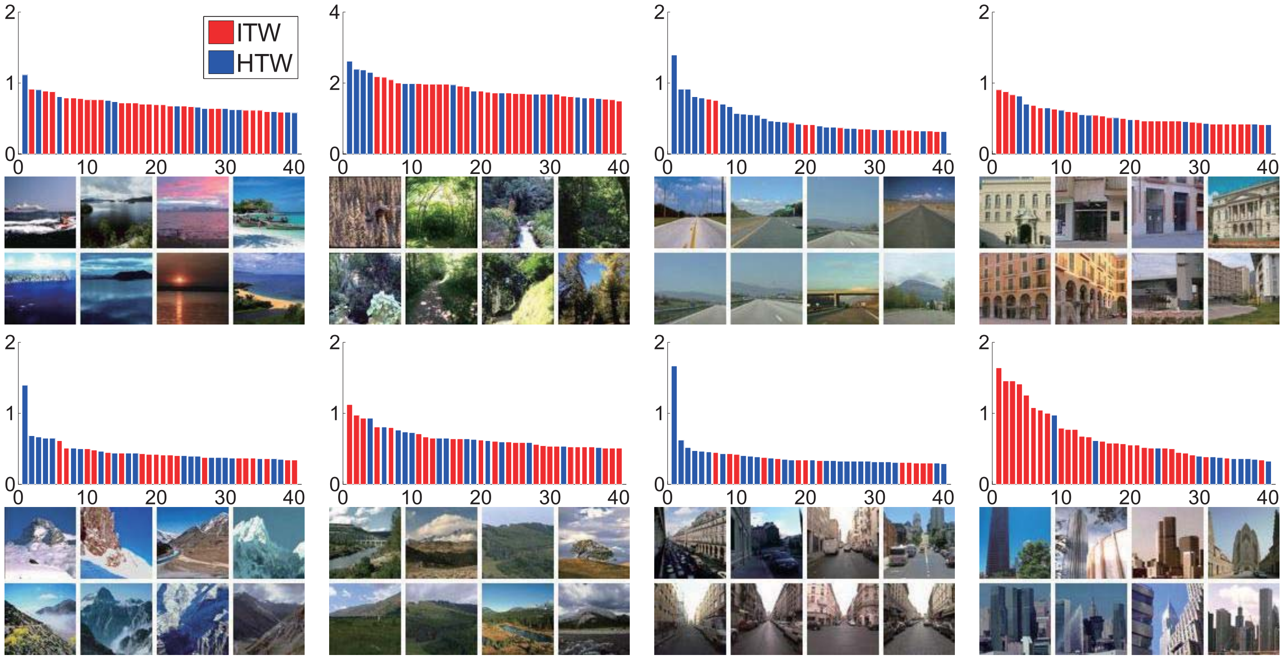

Fig. 4. Selected visual words for 15 categories of the UIUC-Scene database. For each category, we show the top 40 informative visual words according to their weights (the vertical axis). The different colors represent different types of words (i.e., red for ITWs and blue for HTWs).

of categories. The metric purity and conditional entropy are defined as

$$\mathrm{Purity}(X|Y) = \sum_{y \in Y} p(y) \max_{x \in X} p(x|y) \tag{21}$$

$$H(X|Y) = \sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log \frac{1}{p(x|y)} \tag{22}$$

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 images in one category.

### *B. Parameter Settings and Results*

We carry out the experiments on a PC with Quad-Core 3.6 GHz CPU and 32-GB memory. We set the parameter $\beta = 300$ in the probabilistic formulation (8), and the parameter $\tau = 0.2$ in the probabilistic edge definition (5).

In our experiments, we first randomly collect a number of image patches with different scales from the data sets and

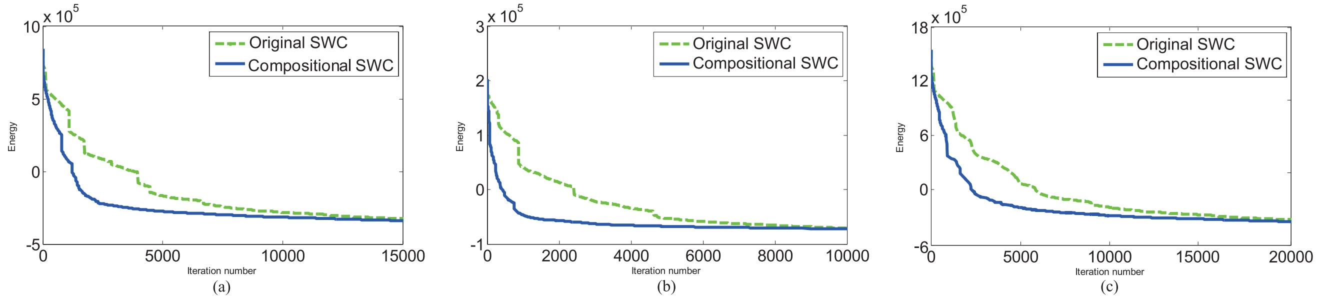

Fig. 5. Convergence comparisons of the CSWC algorithm and the original version. The experiments are executed on the three databases. (a) MIT-Scene, (b) Corel, and (c) UIUC-Scene. In each chart, the horizontal axis and the vertical axis, respectively, represent the iterating step and the target energy ($-\log p(S|\mathcal{D})$). The dashed (green) curves are from the original SWC algorithm and the solid (blue) curves are from the CSWC algorithm, respectively.

generate 500 ITWs and 500 HTWs, as introduced in Section II. There are totally 1000 words in the dictionary.

We carry out our method 10 times and use the average performance for comparison. The inferred category number may not be identical each time, as reported in Table I. The average category number is 9.0 for the MIT-Scene, 10.4 for the Corel, 16.1 for UIUC-Scene, and 25.0 for the mixed data set.

For comparison, several state-of-the-art approaches are implemented based on the codes released by the original researchers, including pLSA [17], affinity propagation (AP) [39] and LDA [40]. For the pLSA approach, we extract color SIFT descriptors to construct a dictionary of 1000 visual words following their original implementation. For the other two approaches, i.e., AP and LDA, we use our image representations (i.e., two types of words extracted within the spatial pyramid) as the inputs of the clustering algorithms. In addition, the $k$-means clustering algorithm is adopted as the baseline, with either our representations or the gradient-based GIST features [6]. These methods use exactly the same experiment settings as our approach for fair evaluation, but the category number for them is manually fixed, i.e., eight for the MIT-Scene database, 10 for the Corel, 15 for the UIUC-Scene, and 23 for the mixed data set. The quantitative performances are reported in Tables II and III based on the two benchmark metrics, respectively. In general, our method outperforms other comparing approaches. We also evaluate our method with only one type of visual words, i.e., either ITW or HTW, so that the benefits of combining two types of features are clearly illustrated.

In our method, the clustering inference is performed simultaneously with the feature selection for category modeling. In Fig. 4, we show the selected visual words of different types, i.e., ITWs and HTWs, for different categories, and the coefficients of top 40 informative words are plotted as well. The results are very reasonable that the selected words match with the appearances of the images very well.

### *C. Analysis*

In the following, we conduct additional empirical analysis to validate the advantages of our approach.

First, we analyze the convergence efficiency of the CSWC algorithm and compare with the original version. Fig. 5 shows the convergence curves of the target energy, i.e., $-\log P(S|\mathcal{D})$, with the increasing iteration steps. Note that the energy goes inversely with the posterior probability. We can observe that the CSWC algorithm converges significantly faster on all the three databases, which is accordant with the result in [34].

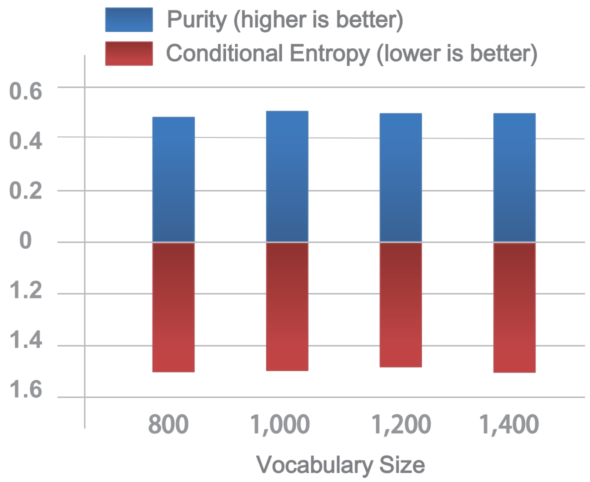

Fig. 6. Influence of vocabulary size. This analysis is executed on the mixed database (of 23 categories). The upper figure and the lower figure, respectively, represent the results via purity and conditional entropy. The horizontal axis represents the vocabulary size. Note that we generate equal size for the two types of words in the testing.

Moreover, we analyze the computational complexity of our approach. The space complexity (i.e., computer memory) is basically related with the size of the visual word dictionary and the number of images to be categorized. Here, we mainly discuss the time complexity that quantifies the amount of time taken by an algorithm conditional on the asymptotic size of the input. Using the big $\mathcal{O}$ notation, which excludes coefficients and lower order terms, the theoretic time complexity of our approach is $\mathcal{O}(MKT)$, where $M$ is the number of sampling steps, $K$ is the category number, and $T$ is the average number of features selected for each category. As we discussed in Section V, the generative model can be pursued analytically by greedy feature selection, and the feature responses on all images can be calculated offline. In addition, only a few (i.e., $<K$) categories need to be updated in each iteration. Hence, we roughly consider the time complexity

determined by the sampling steps. On the mentioned hardware, each iteration costs averagely 0.043 s (MIT-Scene), 0.015 s (Corel), and 0.052 s (UIUC-Scene), respectively, on the three databases.

Finally, to reveal how much the vocabulary size affects the results, we present an experiment in Fig. 6, where the categorization results are reported with different sizes of vocabulary on the mixed data set. The conclusion can be drawn that our approach is not sensitive on the vocabulary size, as we incorporate the model learning (i.e., feature selection) with the categorization. In addition, this property enables us to avoid elaborately tuning the size of vocabulary in practice.

## VII. Conclusion

This paper studies a general framework for automatically discovering image categories via unsupervised graph partition. Compared with the previous methods, the advantage of the proposed method is identified on several public data sets and summarized as follows. First, images are represented by two types of visual words, ITWs and HTWs, which capture image appearances from different aspects. Second, we perform feature selection simultaneously with the clustering procedure, guided by a generative model for each category. Third, we employ a stochastic sampling algorithm for efficient inference, in which the clustering number is automatically determined.

## References

[1] Y. Hu, X. Cheng, L.-T. Chia, X. Xie, D. Rajan, and A.-H. Tan, "Coherent phrase model for efficient image near-duplicate retrieval," *IEEE Trans. Multimedia*, vol. 11, no. 8, pp. 1434–1445, Dec. 2009.

[2] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravì, "Aligning codebooks for near duplicate image detection," in *Multimedia Tools Applications*. New York, NY, USA: Springer-Verlag, 2013, pp. 1–24.

[3] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei, "Building and using a semantivisual image hierarchy," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3336–3343.

[4] X. Zheng, D. Cai, X. F. He, W.-Y. Ma, and X. Y. Lin, "Locality preserving clustering for image database," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, 2004, pp. 885–891.

[5] B. Yao, X. Yang, L. Lin, M. W. Lee, and S. C. Zhu, "I2T: Image parsing to text description," *Proc. IEEE*, vol. 98, no. 8, pp. 1485–1508, Aug. 2010.

[6] 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.

[7] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine, "Unsupervised object discovery: A comparison," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 284–302, 2010.

[8] J. Jeon and R. Manmatha, "Automatic image annotation of news images with large vocabularies and low quality training data," in *Proc. ACM Int. Conf. Multimedia*, 2004.

[9] G. Sheikholeslami, W. Chang, and A. Zhang, "Semantic clustering and querying on heterogeneous features for visual data," in *Proc. 6th ACM Int. Conf. Multimedia*, 1998, pp. 3–12.

[10] D. Dai, T. Wu, and S. C. Zhu, "Discovering scene categories by information projection and cluster sampling," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3455–3462.

[11] L. Walker, W. Renninger, and J. Malik, "When is scene identification just texture recognition?" *Vis. Res.*, vol. 44, no. 19, pp. 2301–2311, 2004.

[12] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE CVPR*, vol. 2. Jun. 2005, pp. 524–531.

[13] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *Proc. 10th IEEE ICCV*, Oct. 2005, pp. 370–377.

[14] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE CVPR*, Jun. 2006, pp. 2169–2178.

[15] S. Battiato, G. M. Farinella, G. Gallo, and D. Ravì, "Scene categorization using bag of textons on spatial hierarchy," in *Proc. ICIP*, Oct. 2008, pp. 2536–2539.

[16] G. M. Farinella and S. Battiato, "Scene classification in compressed and constrained domain," *IET Comput. Vis.*, vol. 5, no. 5, pp. 320–334, Sep. 2011.

[17] A. Bosch, A. Zisserman, and X. Munoz, "Scene classification via pLSA," in *Proc. ECCV*, vol. 3954. Oct. 2006, pp. 517–530.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 993–1022, 2003.

[19] L. Lin, T. Wu, J. Porway, and Z. Xu, "A stochastic graph grammar for compositional object representation and recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1297–1307, 2009.

[20] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[21] G. Carneiro and A. D. Jepson, "Multi-scale phase-based local features," in *Proc. IEEE CVPR*, vol. 1. Jun. 2003, pp. 736–743.

[22] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," in *Proc. 10th IEEE ICCV*, vol. 1. Oct. 2005, pp. 800–807.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE CVPR*, vol. 1. Jun. 2005, pp. 886–893.

[25] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.

[26] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009.

[27] 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.

[28] P. Viola and M. Jones, "Fast multi-view face detection," in *Proc. IEEE CVPR*, Jun. 2003.

[29] 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, Apr. 1997.

[30] Y. Wu, Z. Si, H. Gong, and S.-C. Zhu, "Learning active basis model for object detection and recognition," *Int. J. Comput. Vis.*, vol. 90, no. 2, pp. 198–235, 2010.

[31] D. Liu and T. Chen, "Unsupervised image categorization and object localization using topic models and correspondences between images," in *Proc. ICCV*, 2007, pp. 1–7.

[32] A. Barbu and S.-C. Zhu, "Generalizing Swendsen–Wang for image analysis," *J. Comput. Graph. Statist.*, vol. 16, no. 4, pp. 877–900, 2007.

[33] 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, Aug. 2010.

[34] X. Duan, L. Lin, and H. Chao, "Discovering video shot categories by unsupervised stochastic graph partition," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 167–180, Jan. 2013.

[35] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th CVPR*, Sep./Oct. 2009, pp. 32–39.

[36] 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.

[37] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4844–4857, Dec. 2012.

[38] Z. Si and S. C. Zhu, "Learning hybrid image template (HiT) by information projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1354–1367, Jul. 2012.

[39] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[40] 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, Feb. 2011.

**Liang Lin** (M'12) received the B.S. and Ph.D. degrees from Beijing Institute of Technology, Beijing, China, in 1999 and 2008, respectively.

He was a Joint Ph.D. Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA, from 2006 to 2007, where he was a Post-Doctoral Research Fellow with the Center for Vision, Cognition, Learning, and Art. He is a currently Full Professor with the School of Advanced Computing, Sun Yat-sen University, Guangzhou, China. He has authored more than 50 papers in top tier academic journals and conferences, including PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, *Pattern Recognition*, IEEE Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, *ACM Multimedia*, and the Conference on Neural Information Processing Systems. His research interests include new models, algorithms, and systems for intelligent processing and understanding of visual data, such as images and videos.

Dr. Lin received the Best Paper Runners-Up Award at ACM Symposium on Non-Photorealistic Animation and Rendering in 2010 and the Google Faculty Award in 2012. His Ph.D. dissertation won the China National Excellent Ph.D. Thesis Award Nomination in 2010.

**Ruimao Zhang** received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2011, where he is currently working toward the Ph.D. degree in computer science with the School of Information Science and Technology.

He was a Visiting Ph.D. Student with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2013 to 2014. His research interests include computer vision, pattern recognition, machine learning, and related applications.

**Xiaohua Duan** received the B.B.A degree from the Department of Economic Management, Xi'an University of Posts and Telecommunications, Xi'an, China, in 2004, and the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2012.

His research interests include image/video processing, multimedia analysis and retrieval, and computer vision.