A Hierarchical Visual Model for Video Object Summarization

David Liu, *Member*, *IEEE*, Gang Hua, *Member*, *IEEE*, and Tsuhan Chen, *Fellow*, *IEEE*

Abstract—We propose a novel method for removing irrelevant frames from a video given user-provided frame-level labeling for a very small number of frames. We first hypothesize a number of windows which possibly contain the object of interest, and then determine which window(s) truly contain the object of interest. Our method enjoys several favorable properties. First, compared to approaches where a single descriptor is used to describe a whole frame, each window's feature descriptor has the chance of genuinely describing the object of interest; hence it is less affected by background clutter. Second, by considering the temporal continuity of a video instead of treating frames as independent, we can hypothesize the location of the windows more accurately. Third, by infusing prior knowledge into the patch-level model, we can precisely follow the trajectory of the object of interest. This allows us to largely reduce the number of windows and hence reduce the chance of overfitting the data during learning. We demonstrate the effectiveness of the method by comparing it to several other semi-supervised learning approaches on challenging video clips.

Index Terms—Topic model, probabilistic graphical model, Multiple Instance Learning, semi-supervised learning, object detection, video object summarization.

1 Introduction

The endless streams of videos on the Internet often contain irrelevant data. Our goal is to cut video clips shorter and retain the frames that are relevant to the user input. We assume the user has an "*object of interest*" (OOI) in mind, which can, for example, be a car, a book, or the scene of a forest. The system infers which frames contain the OOI and determines these frames as relevant. This application can be used for, e.g., shortening surveillance videos or TV programs. Fig. 1 shows an illustration.

The amount of user label information, as well as its format, has a major impact on system design. The amount of user label information can range from all frames being labeled to none. For those frames being labeled, the labeling can be as detailed as providing bounding boxes for each frame (which we call pixel-level labeling) or as coarse as "this frame does (or does not) contain the OOI" (which we call frame-level labeling). We consider the case where the system is provided with very limited information. Specifically, the user will label at least one frame as relevant and another frame as irrelevant. These labels are at the frame level instead of at the pixel level. Although pixel-level labeling (such as using a bounding box or segmentation mask to specify the location of the OOI) can provide more information, we intend to explore the possibility of letting the user provide coarser and less tedious labeling.

Frame-level labeling is not only easier to provide, but also can be tightly integrated with real-world applications. For example, when a user watches a video and rewinds to play back a certain portion, the user is implicitly telling the system that she or he is more interested in that part of the video. In other words, the user does not need to consciously do the labeling. The framework we introduce in this paper can readily be applied to such scenarios and takes advantage of such kind of user information.

Our approach is to discover the object of interest in a video given a very small number of frame-level labels. In order to learn the appearance variability of the object of interest, we build up a codebook of local appearances that are characteristic for the object of interest. This is done by extracting local features around interest points and clustering them. Based on this codebook, we simultaneously estimate the appearance, location, and scale of the object. The estimated location and scale are rough estimates, based on which we further sample multiple windows to obtain multiple hypotheses. For the frames that are labeled as positive, we assume that at least one of the hypotheses will genuinely describe the OOI. For the frames that are labeled as negative, we assume that none of the hypotheses correspond to the OOI. Based on these assumptions, we simultaneously learn a window-level classifier and a frame-level classifier. The window-level classifier provides feedback to the location and scale estimates obtained in the previous iteration, using which we can obtain more accurate windows. After a couple of iterations, the estimated windows converge in location and scale, and we use the frame-level classifier to assign a score to each frame that determines the relevant and irrelevant frames.

Our contribution can be summarized as follows: 1) a novel application that summarizes videos based on the

• D. Liu is with Siemens Corporate Research, 755 College Rd. E, Princeton, NJ 08540. E-mail: dawenliu@gmail.com.
• G. Hua is with Nokia Research Center Hollywood, Los Angeles, CA. E-mail: ganghua@gmail.com.
• T. Chen is with the School of Electrical and Computer Engineering, Cornell University, 224 Phillips Hall, Ithaca, NY 14853. E-mail: tsuhan@ece.cornell.edu.

Manuscript received 16 Aug. 2008; revised 15 Dec. 2008; accepted 29 July 2009; published online 21 Jan. 2010.
Recommended for acceptance by Q. Ji, J. Luo, D. Metaxas, A. Torralba, T.S. Huang, and E.B. Sudderth.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2008-08-0535.
Digital Object Identifier no. 10.1109/TPAMI.2010.31.

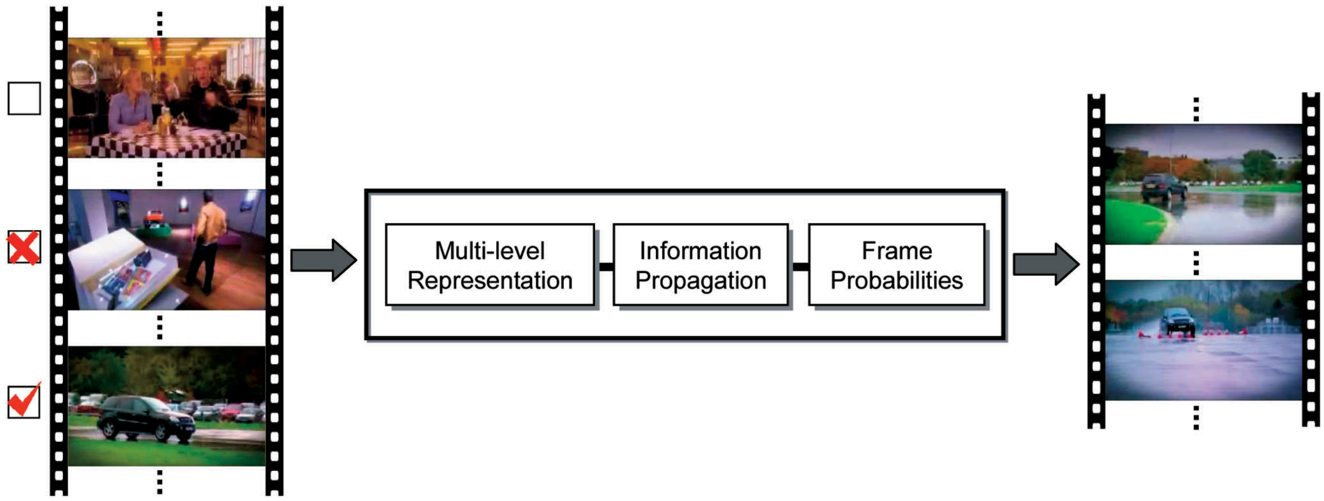


Fig. 1. Video object summarization with frame-level labels. Frames are unlabeled (top left), labeled as irrelevant (middle left), or relevant (bottom left). The system then finds out what the object of interest is (in this case, the black vehicle) and removes the frames that don't contain the vehicle.

implicitly specified OOI, 2) a novel system that uses weakly labeled data for object discovery in video, and 3) a novel method that takes advantage of the temporal smoothness property during semi-supervised learning.

The paper is organized as follows: In Section 2, we give a review on related work. In Section 3, we introduce a multilevel image representation. In Section 4, we explain how the multiple levels interact with each other. In Section 5, we experimentally evaluate the framework and quantify the robustness of the resulting approach in cluttered, low-resolution real-world Internet videos. In Section 6, we conclude the paper with remarks on future directions.

2 Literature Review

This work is an extension of our work in [1], with a more detailed account of the theory and the experiments.

2.1 Extensions of Classic Object Detection

Our approach is deeply connected to the literature on object detection. Classic object detection methods include [2] and [3]. Since then, a lot of work has addressed specific components and drawbacks of these methods. By making comparisons to these specific components, we can provide an overview of the literature and relate them to our work. The specific components of classic object detection methods include at least the following:

1. **The need for pixel-level labeling**. In classic object detection, a bounding box covering the OOI needs to be annotated for each image. In our work, we consider the more challenging task of having as input only frame-level labeling; see Fig. 2 for a comparison. This kind of "weak labeling" is very different from classic object detection, where the characteristics of the OOI are learned from plenty of pixel-level labeled data. It is also different from the recent video retrieval work in [4], [5], where bounding boxes of the OOI are manually specified in a number of frames.

We use the Multiple Instance Learning (MIL) framework [6], which handles the scenario where class labels are associated with sets of samples instead of individual samples, thereby providing a framework for learning based on frame-level labeling. In computer vision problems, there is often no natural choice of samples, and the design of the set of samples varies for different applications. In the context of image categorization [7], class labels are annotated for each image, and the set of samples consists of nonoverlapping windows of 4×4 pixels. In the context of object detection [8], a coarse bounding box is annotated for each image, and the set of samples consists of a large number of windows that are similar in position and scale to the annotated bounding box. The number of windows is a crucial design parameter. If the number of windows is too small and none of the windows truly cover the object

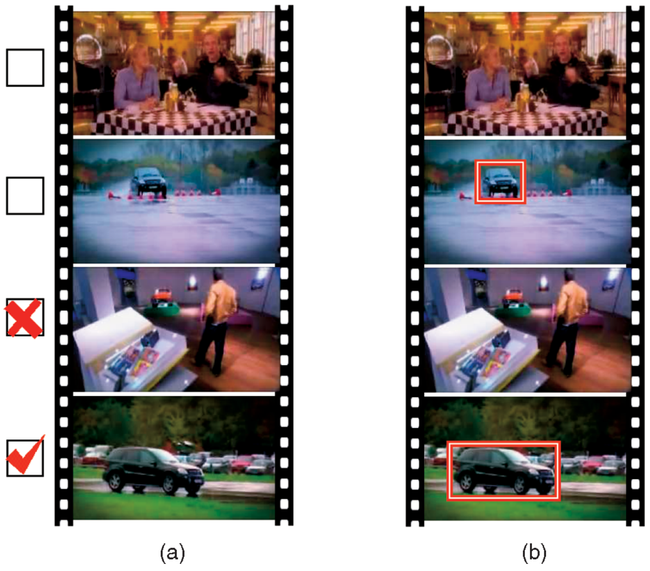


Fig. 2. (a) Labeling at the frame level. (b) The bounding box type of labeling provides more explicit information regarding the object of interest, but is also more tedious in the labeling process.

of interest, the object of interest cannot be learned. A too large number of windows, however, would yield a classifier that is too expressive and cannot generalize properly.

The set of samples in our work is generated differently. We don't need humans to annotate bounding boxes, as in [8]. The windows are automatically sampled from a window-proposal distribution that keeps the number of windows to a small and yet effective number.

2. **Frames are treated independently.** In videos, however, objects tend to move in a smooth trajectory; therefore, the frames should not be treated independently. There are a few object recognition papers that take the temporal smoothness property into account during detection [9], [10], but none of them use frame-level labels during training. Our window-proposal distribution takes into account the temporal smoothness property of videos. Therefore, the windows are "dynamic" and are being sampled differently for different frames.
3. **Detection involves the computation of local features in a fixed spatial configuration.** A fixed spatial configuration is, however, not robust to local variations. One way to model local variations is by representing objects as an assembly of parts or patches. In the Constellation Model [11], an object is represented by a small number of parts, and the joint spatial and appearance distribution are modeled by a fully connected graph that can tolerate slight spatial deformations. In a later version [12], the fully connected graph is replaced by a simpler star topology, which can handle a larger number of parts and take advantage of an efficient inference algorithm [13]. In the Implicit Shape Model [14], the inference of the object position and scale is achieved by a probabilistic extension of the Generalized Hough Transform. In our work, the object appearance is described by a histogram of discretized local feature descriptors, thereby achieving robustness to local variations.

   Different than other part-based methods such as the Constellation Model, the ISM, and the one-shot learning framework [15], we leverage motion consistency to improve detection.

   Many object recognition and detection systems are using a *parts-based* representation [11], [13], [14]. When the scene contains background clutter, parts-based approaches face the difficulty of the grouping problem, that is, which parts should be grouped together to form the object and which parts should be left as background. Another popular representation is the *window-level* representation, commonly used in sliding windows approaches [2], [3], where one considers the statistics inside a window. This approach has been the common practice for face detection [2], [3], and lends itself to weakly supervised learning through the MIL framework [8]. In general, the number of sliding windows that need to be considered is huge, especially when there is no prior knowledge about the location and scale of the object of interest. Under certain circumstances, sliding windows approaches can be vastly accelerated by branch-and-bound techniques [16], but this is beyond our discussion.

   Our image representation is a hierarchical representation that consists of both the part-level *and* the window-level representation. The hierarchical representation provides a bridge between the parts-based approaches and the sliding windows approaches.
4. **Training requires lots of labeled data,** which is not an option in many cases. In some applications, the system needs to learn effectively from limited labeled data. We use self-learning [17], which is a semi-supervised learning framework shown to be useful when labeled data are limited and yet there are plenty of unlabeled data. It incrementally builds a single classifier using labeled training data and converts the most confidently predicted unlabeled data into labeled training examples. It has been used in various domains [18], such as semi-supervised object detection [19], [20]. In [19], learning requires both pixel-level labeled data and frame-level labeled data. An object detector is initially trained on the pixel-level labeled data, and the learned model is used to estimate labels for the frame-level labeled data. Our approach is also based on self-learning, but requires only frame-level labels.
5. **The need for exhaustive window scanning.** An exhaustive search over all possible object locations and scales means high computational complexity, as for an image of as low resolution as 320×240, millions of windows need to be examined. It also imposes constraints on the detector's capability since a large number of potential false positives need to be excluded.

   One novelty of our work is the window-proposal distribution, from which we sample a relatively small number of windows, thereby avoiding exhaustive window scanning.

2.2 Unsupervised and Weakly Supervised Object Detection

Our framework is also related to topic models such as PLSA, LDA, and HLDA [21], [22], [23], [24], [25], [26], [27], [28]. Topic models belong to the family of probabilistic graphical models [29], [30]. Probabilistic graphical models are graphs with nodes representing random variables and edges representing conditional independence assumptions. Topic models have recently been introduced from the text understanding community [31] to computer vision due to their strength in unsupervised learning. We leverage the topic model in [27], [28] and extend it to a semi-supervised setting by incorporating additional prior models during learning. Our image representation, which is based on a hierarchical part-level and window-level representation, is similar to [22] and [26]. The problem solved, the application targeted, as well as the fundamental approach adopted in our paper, are significantly different from these works.

Recent work on weakly supervised learning methods includes [32] and [33]. Chum and Zisserman [33] introduce a method to detect object classes based on frame-level labeling. The region of interest is initially determined by the top N most discriminative patches and iteratively being

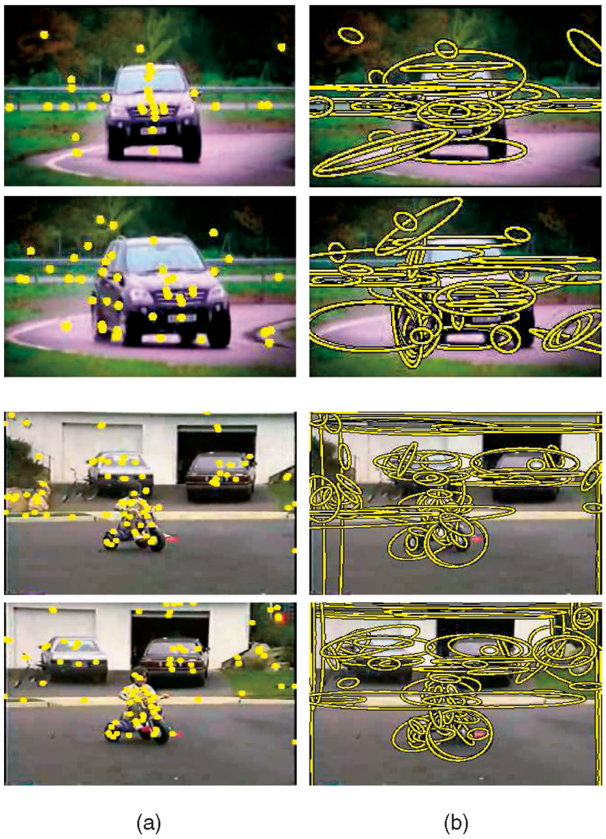


Fig. 3. Maximally Stable Extremal Regions (MSERs). (a) Position of MSERs. (b) Coverage of MSERs.

updated. Finally, a classifier is trained based on the regions of interest. Our approach is model-based and therefore estimates the region of interest in a more principled manner. It also avoids hard thresholding of the patches.

2.3 Object Instance Recognition and Retrieval

Object instance recognition in [34], [4], and [5] is matching-based approaches; in particular, Sivic et al. [5] take into account temporal information during matching, and their study is related to the literature of wide-baseline matching in multiple-view geometry. Our method is based on a probabilistic model and is related to topic models and Multiple Instance Learning. Our model considers the image as being composed of foreground and background, and learning takes advantage of both the positive and negative frames. Matching-based methods generally do not explicitly make use of image background or negative frames.

The vast literature on retrieval [4] and relevance feedback [35] is related to our work. Image retrieval systems often allow users to provide positive and negative feedback; hence, the task of image retrieval can also be cast under the self-training [17] or Multiple Instance Learning framework [8], [7]. Nonetheless, our system exploits temporal information of videos in a novel way which distinguishes itself from the image retrieval literature.

2.4 Contextual Object Detection and Other Work

It has been observed that background context is often correlated with the foreground object, and hence modeling their relationship can benefit object detection [36], [37], [38].

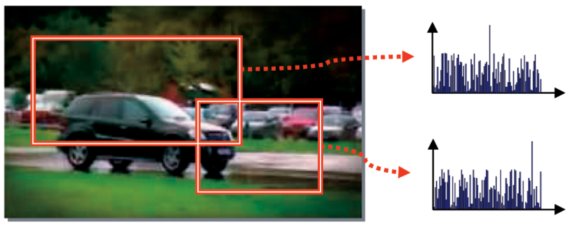


Fig. 4. Windows, each represented by a histogram over visual words. We use a variety windows with different locations and scales.

In our application, the location of the object of interest is not labeled. The topic model models the foreground and background histograms and hence takes into account background context. The Multiple Instance Learning framework uses multiple windows that hypothetically cover the object of interest and automatically determines the window size. These windows also naturally include a certain amount of background context.

Our hierarchical visual model propagates information across multiple levels of image representations, namely, the patch level, the window level, and the frame level. The concept of hierarchically propagating information across layers has been used in the object recognition literature for building up complex detectors from smaller ones [39], [40], for learning atomic parts of object structure and parsing image compositions [41], [42], [43], [44], [45], and for hierarchical decomposition of the set of possible presentations of an object [46].

In [47], activities in a video are condensed into a shorter period by simultaneously showing multiple activities. It does not intend to *discover* the frames that contain the user-desired OOI from limited user input.

3 Multilevel Image Representation

3.1 Patch-Level Representation

We use the Maximally Stable Extremal Regions (MSER) operator [48] to extract *patches* or *parts* (we use these two terms interchangeably) from video frames.[1] Examples are shown in Fig. 3. MSERs are the patches of an image where local contrast is high. Other operators could also be used; see [49] for a collection.

Patch-level features are extracted from these MSERs by Scale Invariant Feature Transform (SIFT) [34]. In this work, we extract MSERs and SIFT descriptors from gray-scale images, although color images [50] can also be used. The SIFT features are collected from all frames in the video and are vector quantized using K-Means Clustering. The resulting $J = 50$ cluster centers form the dictionary of visual words, $\{w_1, \ldots, w_J\}$. Each patch is then represented by its closest visual word in the SIFT feature space.

3.2 Window-Level Representation

A video frame is covered by multiple windows, each window being a hypothesis trying to explain the location and scale of the OOI. A window is represented by a histogram of *visual words*, or *textons* [51], as shown in Fig. 4.

1. The word "region" in MSER should not be confused with the "windows" to be introduced later. Each window consists of a set of MSERs.

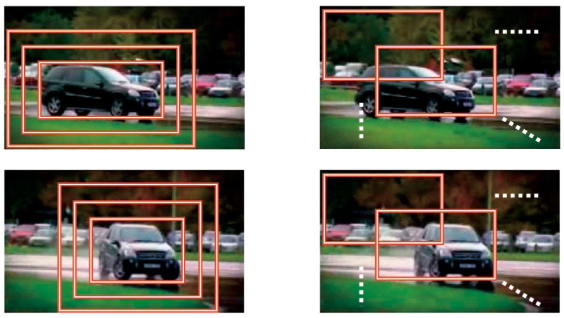


Fig. 5. Window-proposal distribution. Left: Informative Windows. Right: Random Windows.

Windows are sampled from a *window-proposal distribution*, defined as a probability mixture model,

$$p_k^{\mathrm{WP}}(\mathbf{r}) = \alpha\mathcal{N}(\mathbf{r}|\hat{\mathbf{r}}_\mathbf{k}, \hat{\mathbf{\Sigma}}_k) + (1-\alpha)\mathcal{N}(\mathbf{r}|\hat{\mathbf{r}}', \hat{\mathbf{\Sigma}}'), \quad (1)$$

where k is the frame index.

The first distribution, $\mathcal{N}(\mathbf{r}|\hat{\mathbf{r}}_\mathbf{k}, \hat{\mathbf{\Sigma}}_k)$, is an estimate of where the object of interest is located at. The derivation of the parameters is deferred until Section 4.2.

The second mixture component is a Gaussian-near-uniform distribution, and the purpose of including it is to increase the robustness when the first mixture component is estimated poorly. The parameter $\hat{\mathbf{r}}'$ is set to the center of the image plane, and $\hat{\mathbf{\Sigma}}'$ is set with large diagonal variances equal to $1{,}000^2$. Such types of outlier distributions have been used in the real-valued graphical models literature [52].

To sample a window, we first select one of the two distributions according to the value of α, and then sample a point from the respective Normal distribution. This gives us the center of the window. Based on the center, we use a 1.2 scale ratio to obtain different window sizes, with the smallest one equal to the variance specified by $\hat{\mathbf{\Sigma}}_k$. This is illustrated in Fig. 5, with the informative windows sampled from the first mixture component and the random windows sampled from the second component. Using various sizes is to increase the robustness in case of inaccurate scale estimates.

Finally, each window is represented by a histogram, $x_{k,n}$, of visual words that it contains, where n is an index over windows.

This window sampling scheme yields informative windows, to be distinguished from the common practice of using uniform sampling or uninformative windows in Multiple Instance Learning [7]. Besides, the window-proposal distribution is frame dependent and hence the windows are "dynamic," i.e., the windows change their locations and scales over frames.

4 Information Propagation between Levels

The window-level representation provides a bridge between frame-level labels and patch-level observations. In the following sections, we explain how information is propagated between these levels. Basically, in the first iteration, we sample windows from a near-uniform distribution. Using the framework in Section 4.1, we propagate user label information from frames to windows. A window classifier is learned, which assigns patch scores. Using the framework in Section 4.2, an updated spatial distribution is estimated. Starting from the second iteration, we sample windows based on the updated spatial distribution.

4.1 Information Propagation between Window Level and Frame Level

Given the multiple windows, our goal in this section is to propagate the information from the frame-level labels into the windows. In other words, we will estimate how likely it is that each window contains the OOI. Formally, each frame has a *frame probability* q_k and each window has a *window probability* $q_{k,n}$. The *frame probability* is the probability that the frame contains the OOI and the *window probability* is the probability that the window contains the OOI. This section explains how to estimate these probabilities.

Assume a frame is labeled by the user as positive when at least one of its windows covers the OOI and negative when none of its windows cover the OOI. This assumption naturally corresponds to the Noisy-OR model [29], which models the relationship between q_k and $q_{k,n}$ as follows:

$$q_k = 1 - \prod_{n=1}^{N_k}(1 - q_{k,n}), \quad (2)$$

where N_k is the number of windows in frame k. We can see that the frame probability is close to zero if and only if the window probabilities are all close to zero. We also see that once $q_{k,n}$ is determined, so is q_k. The MILBoost [8] and AnyBoost [53] algorithms provide a way to estimate $q_{k,n}$ as follows:

Define a *window classifier* that has the form of a weighted sum of weak learners,

$$C(.) = \sum_j \lambda_j c_j(.), \quad c_j(.) \in \{-1, +1\}. \quad (3)$$

Assume that the window probability is related to the output of the window classifier, $C(x_{k,n})$, through the logistic sigmoid function,

$$q_{k,n} = 1/(1 + exp(-C(x_{k,n}))). \quad (4)$$

The goal now is to estimate $\{\lambda_j\}$ and $\{c_j(.)\}$ so that $q_{k,n}$ approaches its true value. Since $C(.)$ is additive, this can be achieved by maximizing likelihood through gradient ascent [53]. Here, we outline the procedure. Let $u_k \in \{0, 1\}$ denote the label of frame k. The likelihood is defined as

$$L = \prod_k {q_k}^{u_k}(1 - q_k)^{(1-u_k)}, \quad (5)$$

and is maximum when $q_k = u_k$. To find a window classifier that maximizes the likelihood, we first express the log-likelihood as a function of $C(.)$. The derivative of the log-likelihood with respect to $C(.)$ is easily derived as $\frac{u_k - q_k}{q_k} q_{k,n}$, which we denote by $\varpi_{k,n}$. Gradient ascent in each round j is then achieved in two steps. First, one solves the optimal $c_j(.) \in \{-1, +1\}$ through the optimization problem

$$c_j(.) = \arg\max_{c'(.)} \sum_{k,n} c'(x_{k,n})\varpi_{k,n}. \quad (6)$$

Second, a line search is performed to seek for the optimal λ_j, i.e.,

$$\lambda_j = \arg\max_{\lambda} L(C + \lambda c_j). \quad (7)$$

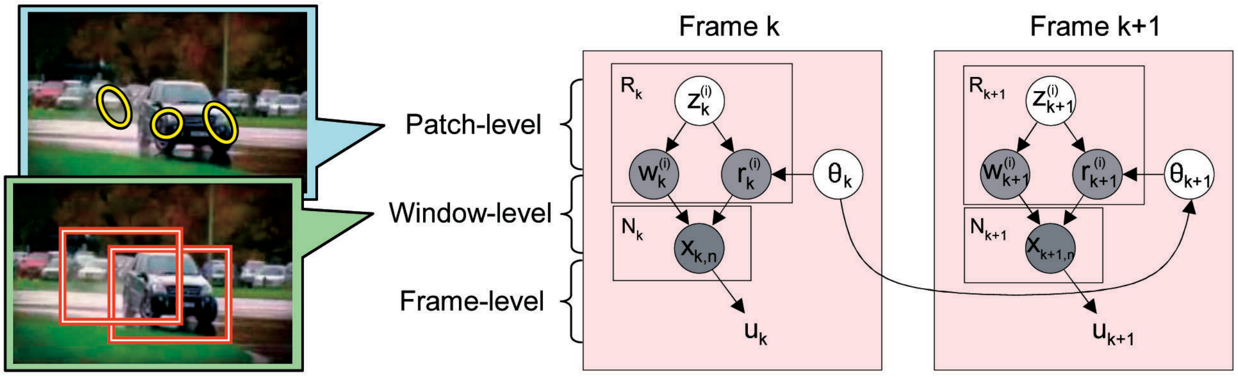


Fig. 6. The graphical model in plate notation. R_k is the number of patches and N_k is the number of windows in frame k. Each patch is associated with a visual word w, a location r, and a hidden variable z. The histogram of visual words inside a window is denoted as x. Frame labels are denoted as u, which are observed for the labeled frames and hidden for the unlabeled frames. The location and scale parameters of the window-proposal distribution are denoted as θ.

After these two steps, $C(.)$ is updated by $C(.) \leftarrow C(.) + \lambda_j c_j(.)$, and a new round of gradient ascent begins.

As a summary, the boosted classifier is operating at the window level, but the labels are only provided at the frame level, and (2) and (5) provide the links between the two.

Since only a subset of frames are labeled, we use a self-training procedure as follows: First, we use the user-labeled frames to train a classifier $C(.)$ as explained above. The classifier is then trained again with the pseudolabels included, which are obtained by classifying the unlabeled frames. We call the resulting classifier the Self-learning-MILBoost classifier, or S-MILBoost for short. At the end, S-MILBoost estimates the frame probabilities and the window probabilities. The frame probabilities are used to classify the frames into positives and negatives. The window probabilities will be utilized in the next section.

4.2 Information Propagation between Patch Level and Window Level

Now that we have derived the frame probabilities, we can classify the frames into those that contain the OOI and those that don't. However, the derivation is based on the premise that at least one of the windows reasonably covers the OOI. If this assumption doesn't hold, the Noisy-OR model would be invalid, and the classifier would be biased. In this section, we introduce a way to estimate where the OOI is located based on information propagation between patches and windows. This extra layer of propagation introduces new information and provides a good estimate of the OOI location.

The hidden variable z denotes the origination of a patch, being either background, z_-, or object of interest, z_+. We assert that the joint distribution of visual word w, location r, and hidden label z of a patch in frame k obeys the conditional independence assertions postulated in the probabilistic graphical model in Fig. 6. This corresponds to the following decomposition:

$$p_k(z,r,w) = p_k(r|w,z)p_k(w|z)p_k(z) \equiv p_k(r|z)p(w|z)p_k(z). \quad (8)$$

The distribution $p_k(z)$ gives the prior probabilities of a patch in frame k originating from the OOI and from the background. When $p_k(z_+)$ is very small, the OOI is very unlikely to be present in frame k. This means we can model occlusion and disappearing objects.

The appearance distribution $p(w|z_+)$ models the appearance of the OOI. Likewise, $p(w|z_-)$ models the appearance of the background. In (8), we assert that $p_k(w|z) = p(w|z)$, which means the appearance distribution is frame independent. This is the same assertion made in the PLSA model [31], [21]. This assertion allows us to learn the object appearance using all frames.

The spatial distribution, $p_k(r|z)$, describes how the patches from the OOI and those from the background are distributed.

We use the Expectation-Maximization (EM) algorithm [54] to estimate the distributions $p_k(z)$, $p(w|z)$, and $p_k(r|z)$. The EM algorithm consists of two steps. The E-step computes the posterior probabilities for the hidden variables:

$$p_k\big(z_k^{(i)}|w_k^{(i)},r_k^{(i)}\big) = \frac{p_k\big(z_k^{(i)}\big)p\big(w_k^{(i)}|z_k^{(i)}\big)p_k\big(r_k^{(i)}|z_k^{(i)}\big)}{\sum_{z\in\{z_+,z_-\}} p_k(z)p\big(w_k^{(i)}|z\big)p_k\big(r_k^{(i)}|z\big)}, \quad (9)$$

where i is an index over all patches in frame k, and $z_k^{(i)} \in \{z_+, z_-\}$ denotes the hidden origination of patch i. In the M-step, we adopt a Bayesian approach to estimating the probabilities, using m-probability estimation [55]:

$$p_k(z_+) = \frac{\sum_{i=1}^{R_k}\big(m_k^{(i)} + p_k\big(z_k^{(i)} = z_+|w_k^{(i)},r_k^{(i)}\big)\big)}{2R_k}, \quad (10)$$

$$p_k(z_-) = \frac{\sum_{i=1}^{R_k}\big(1 - m_k^{(i)} + p_k\big(z_k^{(i)} = z_-|w_k^{(i)},r_k^{(i)}\big)\big)}{2R_k}, \quad (11)$$

where R_k is the number of patches in frame k and $m_k^{(i)}$ is prior information of patches obtained as follows: Based on the window probability of S-MILBoost, we define the *patch score* as an indicator of how likely a patch originates from the OOI. Noticing that each patch can belong to multiple windows, we define the patch score as the largest window probability among the windows that cover the patch. The patch score is denoted by $m_k^{(i)}$, where i is indexing over all patches in frame k.

The appearance distribution is updated as follows:

$$p(w'|z_+) = \frac{1}{b}\sum_{k,i}\big(m_k^{(i)} + p_k\big(z_k^{(i)} = z_+|w_k^{(i)} = w', r_k^{(i)}\big)\big), \quad (12)$$

$$p(w'|z_-) = \frac{1}{b'} \sum_{k,i} \left(1 - m_k^{(i)} + p_k\left(z_k^{(i)} = z_- | w_k^{(i)} = w', r_k^{(i)}\right)\right), \tag{13}$$

where k is summing over all frames, i is summing over all patches in frame k, and b and b' are normalization constants so that $p(w|z_+)$ and $p(w|z_-)$ are proper probability distributions. The above update equations are derived in the same manner as in [31], with the addition of patch scores for the m-probability estimation.

4.2.1 Temporal Smoothing

The spatial distribution is derived in a manner that takes into account the temporal smoothness of object motion. Intuitively, we use a filter that smoothes the estimated trajectory of the OOI. While the Kalman filter models the trajectory of only a single spatial-temporal observation, the PDA filter [56] explicitly models target-originated and background-originated spatial-temporal observations; hence we use the PDA filter as described below.

Define the state $\mathbf{s}_k$ as the unknown location and velocity of the OOI in frame k. We assume a constant velocity motion model and the state evolves according to $\mathbf{s}_{k+1} = \mathbf{F}\mathbf{s}_k + \xi_k$, where $\mathbf{F}$ is the state matrix and the process noise sequence ξ_k is white Gaussian. If patch i in frame k originates from the OOI, then its location can be expressed as $\mathbf{r}_k^{(i)} = \mathbf{H}\mathbf{s}_k + \zeta_k^{(i)}$, where $\mathbf{H}$ is the output matrix and the observation noise sequence $\zeta_k^{(i)}$ is white Gaussian; otherwise, the location is modeled as a uniform spatial distribution.[2]

The state estimate can be written as $\hat{\mathbf{s}}_k = \sum_{i=1}^{R_k} \hat{\mathbf{s}}_k^{(i)} \beta_k^{(i)}$, where $\hat{\mathbf{s}}_k^{(i)} = \hat{\mathbf{s}}_{k^-} + \mathbf{W}_k \epsilon_k^{(i)}$ is the updated state estimate conditioned on the event that patch i in frame k originates from the OOI, where $\epsilon_k^{(i)} = \mathbf{r}_k^{(i)} - \hat{\mathbf{r}}_{k^-}$ is the innovation, $\hat{\mathbf{r}}_{k^-}$ is the observation prediction, $\hat{\mathbf{s}}_{k^-}$ is the state prediction, and $\mathbf{W}_k$ is the Kalman Filter gain [56]. The state estimation equations are essentially the same as in the PDA filter. The association probability $\beta_k^{(i)}$ is defined as $\beta_k^{(i)} \propto \mathcal{N}(\epsilon_k^{(i)}|0, \boldsymbol{\Upsilon}_k) p_k(z_k^{(i)}|w_k^{(i)}, r_k^{(i)})$, where the first term contains motion information, the second term contains appearance, location, and scale information, and $\boldsymbol{\Upsilon}_k$ is the innovation covariance. We then have

$$p_k\left(r_k^{(i)}|z_+\right) = \frac{1}{b} \mathcal{N}\left(\mathbf{r}_k^{(i)}|\hat{\mathbf{r}}_k, \hat{\boldsymbol{\Sigma}}_k\right), i = 1, \ldots, R_k, \tag{14}$$

where $\hat{\mathbf{r}}_k = \mathbf{H}\hat{\mathbf{s}}_\mathbf{k}$ is the location estimate and b is a constant to ensure the spatial distribution is normalized. The spatial distribution, $p_k(r|z_+)$, is a probability mass function, taking unnormalized values from the Normal distribution in (14). The *weighted* covariance matrix $\hat{\boldsymbol{\Sigma}}_k$ is the covariance matrix of the locations $\mathbf{r}_k^{(i)}$ with a weighted mass for each data point, with weights equal to the association probabilities $\beta_k^{(i)}$. As a result, if the association probabilities have high uncertainty, the spatial distribution will be flatter; if low uncertainty, it will be sharper around the location of the OOI. The spatial distribution of background patches, $p_k(r|z_-)$, is assumed to be a uniform distribution.

2. With an abuse of notation, $\mathbf{r}$ denotes the continuous image plane coordinates, while r denotes the finite set of patch locations with cardinality R_k for frame k.

We use $\theta_k = (\hat{\mathbf{r}}_k, \hat{\boldsymbol{\Sigma}}_k)$ to denote the parameters of the spatial distribution. As illustrated in Fig. 6, θ is correlated across frames due to the motion filtering framework explained above. This allows us to learn the appearance, location, and scale information of the OOI using all frames, while taking advantage of the motion smoothness property of objects in videos.

A similar framework appears in [27], [28]. We improve it by using prior knowledge, indirectly learned from the frame-level user labels. More specifically, the window probabilities in S-MILBoost contain information about the OOI. This prior knowledge is incorporated into the estimation of the appearance and spatial distributions in (9)-(14). On the other hand, the framework in [27], [28] was completely unsupervised.

4.3 Summary

In the first iteration, we sample windows with $\alpha = 0$ in (1), i.e., sample only from the Gaussian-near-uniform distribution. Using the framework in Section 4.1, we propagate user label information from frames to windows. A window classifier is learned which assigns patch scores. Using the framework in Section 4.2, an updated spatial distribution, $\mathcal{N}(\mathbf{r}|\hat{\mathbf{r}}_\mathbf{k}, \hat{\boldsymbol{\Sigma}}_k)$, is estimated. Starting from the second iteration, we sample windows with $\alpha = 0.5$ in (1). We notice that the system usually converges after three iterations.

5 Experiments

We use 15 video clips from YouTube.com and TRECVID [57]. Sample frames are shown in Fig. 7. Most of the clips are commercial advertisements with a well-defined OOI and range from 20 to 356 seconds in length. We sample each video at two frames per second. In total, there are 3,128 frames of size 320×240. The frames have visible compression artifacts.

The video frames are ground-truthed as positive or negative according to whether they contain the OOI determined by three human labelers, e.g., in the soda video, the soda logo is the OOI. Each method is run 30 times (due to the stochastic nature of the k-means visual word clustering and the EM algorithm), where in each run we randomly select N_p frames from the positive frames and N_n frames from the negative frames as labeled data, where N_p and N_n are one or three. The rest of the frames are treated as unlabeled data. Some videos only have four positive frames; therefore, we used up to $N_p = 3$. Results are averaged over 20 runs. The labeled frames are labeled at the frame level but not pixel level.

Table 1 shows the average precision of different methods. We compare our framework with several other methods that replace or omit some modules of the proposed method. We also compare with some important prior work in literature. The methods are first characterized by the type of annotation available during training (frame level versus window level). The methods are further characterized by frame-based versus window-based image representation. In frame-based representation, features are extracted from the whole frame; in window-based representation, features are extracted from one or multiple windows. In the following, we will introduce these comparative methods while we

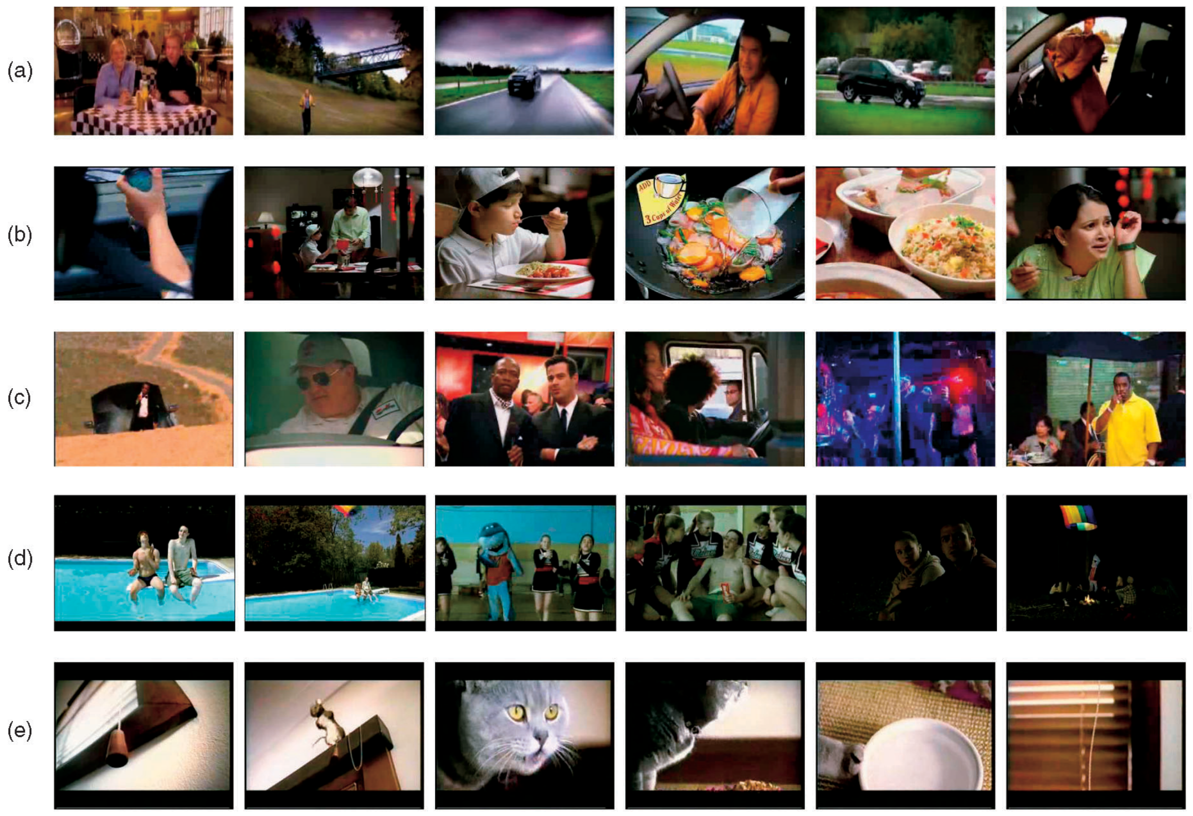


Fig. 7. Sample frames. Name of video clip: (a) car, (b) spice, (c) sode, (d) candy, and (e) cat food.

discuss the results. Some results are not listed in the table but only mentioned in the text.

Method-A: Frame-level learning. The histograms of the labeled frames along with their labels are used to train a classifier. The classifier we use is Discrete AdaBoost [58]. The classifier is then applied to the unlabeled frames. Frames with high confidence scores are assigned pseudolabels. The pseudolabeled data are combined with the original labeled data and the classifier is trained again. This kind of self-training [17] procedure has been used extensively in different domains [18], [19] and achieved top results in the NIPS competition [59].

Method-A learns at the frame-level representation. There are two disadvantages to learning at the frame level:

1. The OOI can be small and visual words from the whole frame are usually dominated by background clutter. Hence, the full-frame histogram representation is more of a representation of the background clutter, instead of representing the OOI.
2. Objects in video often follow a smooth motion trajectory. When working with frames instead of with windows, this smoothness property cannot be readily exploited.

Methods-B and C described below learn at the window-level representation and significantly improve in performance.

Method-B: Window-level learning with random windows. Method-B uses S-MILBoost as discussed in Section 4.1 to propagate information between the frame level and the window level. This method outperforms Method-A in most cases. Using windows helps the learning process to focus on features originated from the OOI and get less affected by background clutter.

Method-B is a simplified version of our final proposed method, Method-C. It has the S-MILBoost component, but it does not learn a window-proposal distribution. Instead, the windows are placed in a fixed pattern as follows: The windows consist of rectangles of size 160×120 with equal spacing between each other. In addition, a rectangle of size 320×240 covering the whole frame is used in order to take care of large objects. After running S-MILBoost, we do not refine the placing of windows, as we do in the proposed method.

We experimented with different numbers of rectangles by changing the spacing between them and obtained different performances, as shown in Fig. 8. There is a sweet spot at the number of 10 windows, which shows that having more windows does not necessarily yield better performance. Even though increasing the number of windows will increase the chance that one of the windows faithfully represents the OOI, the Noisy-OR Multiple Instance Learning framework will become too expressive and generalize worse, hence the drop in performance. We also experimented with placing the windows more concentrated around the center of the frame but obtained similar results.

Method-B′: Most confident window propagation. This method is not listed in the table. It is an extension of Method-B, and is the closest one to the proposed method. See illustration in Fig. 9. Windows are first placed regularly as in Method-B. The window classifier in Section 4.1 is then applied to the windows in the labeled frames, and the window with the highest score in each frame is called the

TABLE 1
Comparing the Average Precision (Percent)

Method ID			**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**
	Annotation		Frame					Window		
	Representation		Frame	Window			Frame	Window		
Sequence	Annot	Chance	Ada	MIL	MIL	Chum	Lowe	Lowe	Ada	MIL
Car*	1+,1-	32.6	58.6	60.6	60.6	39.2	39.0	40.1	32.5	60.6
	3+,3-	32.5	48.8	51.7	49.2	43.1	42.8	46.8	41.3	53.2
Soda*	1+,1-	34.1	68.6	71.1	71.9	73.6	62.4	63.0	65.7	73.0
	3+,3-	33.7	86.7	79.0	83.9	81.5	77.5	72.7	88.7	84.1
CatFood*	1+,1-	43.7	86.1	89.0	91.8	93.1	84.5	84.5	90.4	94.0
	3+,3-	43.5	88.3	90.4	92.6	94.7	90.6	92.3	95.0	94.5
Candy*	1+,1-	3.9	26.9	20.2	20.2	35.6	31.0	31.7	31.4	27.8
	3+,3-	2.0	42.9	57.9	57.9	54.2	48.2	57.0	61.5	62.4
CleanClear	1+,1-	21.2	69.3	87.7	90.3	95.2	81.6	79.3	64.6	88.7
	3+,3-	19.4	99.9	100.0	99.3	100.0	99.8	98.3	98.9	100.0
Cat	1+,1-	39.0	68.1	84.4	82.7	71.4	99.9	100.0	82.1	85.0
	3+,3-	38.2	86.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0
E-Aji*	1+,1-	27.1	24.2	43.0	43.2	30.1	35.2	29.4	28.5	46.1
	3+,3-	25.5	59.2	59.5	62.8	59.3	52.6	43.6	58.7	64.1
CaramelNut*	1+,1-	25.9	37	42.7	41.6	39.6	42.1	49.1	30.0	42.5
	3+,3-	24.1	50.9	55.8	61.1	50.3	56.9	70.7	38.1	63.9
Spice*	1+,1-	20.7	84.4	85.4	85.8	86.1	80.9	80.1	78.7	85.4
	3+,3-	18.5	79.8	89.4	88.5	85.6	79.2	81.0	79.4	88.5
Kellogs	1+,1-	18.4	96	98.9	98.9	90.4	98.5	95.1	95.4	96.1
	3+,3-	14.7	94.1	100.0	100.0	99.9	100.0	100.0	100.0	100.0
FlightSimul*	1+,1-	10.8	21.3	28.3	28.3	31.5	18.4	18.9	21.6	28.5
	3+,3-	10.5	33.8	36.8	62.1	50.3	36.6	45.1	32.7	64.2
SpaceShuttle*	1+,1-	4.8	2.8	3.5	4.2	4.2	3.7	4.1	3.3	4.5
	3+,3-	4.2	12.7	27.7	25.1	27.3	18.2	24.5	19.4	25.4
WeightAero*	1+,1-	11.6	38.1	27.8	44.9	35.3	28.5	25.6	31.2	48.1
	3+,3-	11.2	56.3	40.9	56.1	49.9	30.6	27.1	62.4	59.3
WindTunnel*	1+,1-	24.1	15.0	36.1	35.2	25.2	22.6	26.1	14.2	28.4
	3+,3-	23.8	47.2	56.9	56.1	33.6	60.3	58.5	39.9	53.8
Horizon*	1+,1-	11.2	18.4	22.6	34.1	22.1	29.3	31.4	17.7	36.1
	3+,3-	10.5	41.3	44.1	54.6	37.2	41.8	44.7	36.0	54.6
Average	1+,1-	**21.9**	**47.7**	**53.4**	**55.6**	**51.5**	**50.5**	**50.6**	**45.8**	**56.3**
	3+,3-	**20.8**	**61.9**	**66.0**	**70.0**	**64.5**	**62.3**	**64.2**	**63.5**	**71.2**
Average*	1+,1-	**20.9**	**40.1**	**44.2**	**46.8**	**43.0**	**39.8**	**40.3**	**37.1**	**47.9**
	3+,3-	**20.0**	**54.0**	**57.5**	**62.5**	**55.6**	**52.9**	**55.3**	**54.4**	**64.0**

The number of labeled frames are one positive (1+) and one negative (1−) in the upper row, and three positives and three negatives in the lower row for each video sequence. Method-C is the proposed method. Videos with an (*) are the more difficult ones and have a separate average precision at the bottom.

"base window." Each unlabeled frame then obtains an additional window by interpolating the location of the base windows from the nearest labeled frames. Nearness can be defined as the visual similarity between frames or as the time difference between frames. We found the latter to work better. The replicated window is then replicated again within the frame with different sizes using a 1.2 scale ratio between two windows, with the smallest one equal to the size of the base window and no more than five windows in total. Since videos often contain multiple scene transitions or shots, we only allow the replication to happen within a shot and not across shots. The average precision of this method is 53.9 and 67.3 percent for $(1+, 1-)$ and $(3+, 3-)$, respectively.

Method-B′ is different from the final proposed method in that there is no information propagating between the patch

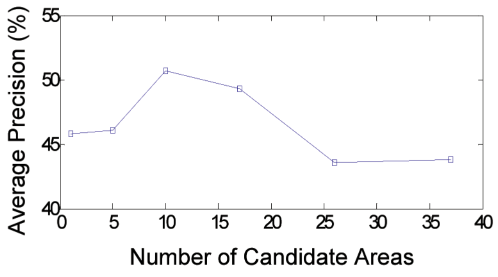


Fig. 8. Increasing the number of windows does not lead to increase in performance.

level and the window level. Therefore, its windows often don't cover the OOI as well as in the proposed method.

Method-C is the proposed method. From Table 1, we see that Method-B and Method-C are most of the time the best performers when using frame-level labels. Together with Fig. 8, this justifies our expectation that properly placed windows are crucial to the performance. Using a huge number of windows overfits the data and lowers the performance. Using a proper window-proposal distribution reduces the need for a large number of uninformative windows. In Fig. 10, we display some frames that are inferred by the proposed method as positive, together with the window with highest window probability shown in yellow. Processing speed is around 2 seconds per frame for patch-level feature extraction, and 1 second per frame for the rest of computation.

There are some failure cases. In the soda video (Fig. 7c), frames that contain the soda logo are labeled as positive frames. For this video, Method-C performs slightly worse than Method-B′. The reason is that the spatial distribution in (14) is centered around the truck front in some frames, and the window-proposal distribution in turn generates many windows that cover the truck front. This increases the chance that the system considers the truck front as the OOI, instead of the logo. One such frame is shown in Fig. 11. Another example is shown for the candy video. Frames containing the candy logo are labeled as positive. Performance of using random windows (Method-B) is on par with using informative windows (Method-C). We noticed that the spatial distribution in Method-C does not truly cover the OOI. One such frame is shown in Fig. 11.

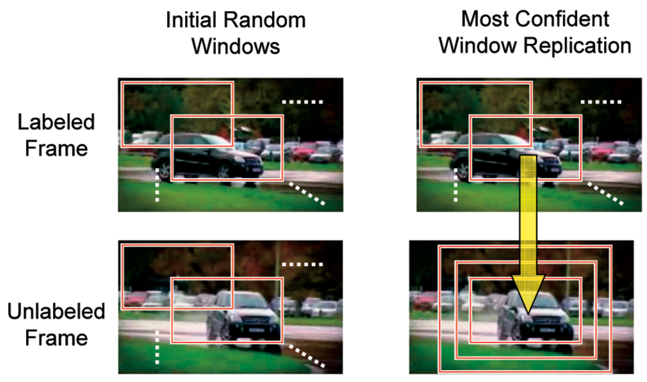


Fig. 9. Illustration of Method-B'.

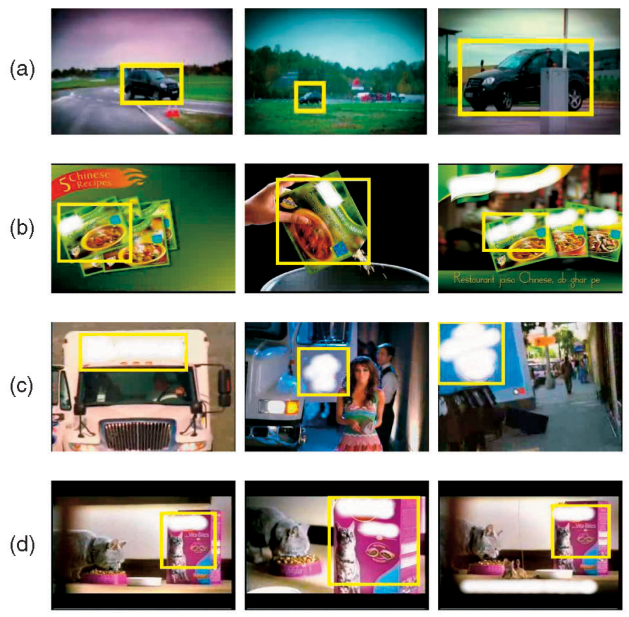


Fig. 10. Sample frames that are inferred as positive. A yellow box shows the window with highest window probability. Name of video clip: (a) car, (b) spice, (c) soda, and (d) cat food.

Method-D. This is the method presented in [33]. We use the patch-level representation as described in Section 3.1. The method tries to identify the region of interest iteratively, starting from the region identified by the 10 most discriminative visual words. Varying this number beyond 15 lowers the performance. Finally, Discrete Adaboost is trained based on the regions of interest. While this method is similar to Method-C in that it tries to estimate the location of the OOI, it does not maintain multiple windows with varying amount of background context as in Method-C, and it also does not model the motion smoothness. Therefore, its performance is closer to Method-B and worse than Method-C. Besides, Method-C estimates the region of interest based on a probabilistic model and avoids hard thresholding of the patches.

Method-E. Using Lowe's method [34] to match SIFT descriptors from an unlabeled frame to a positive frame. Good matches are identified by finding the two nearest neighbors of each SIFT descriptors from the first image among those in the second image, and only accepting a match if the distance to the closest neighbor is less than 0.6 of that to the second closest neighbor. The score of an unlabeled frame is the number of good matches found. When multiple positive frames are available, we keep the largest score. We made the method a semisupervised

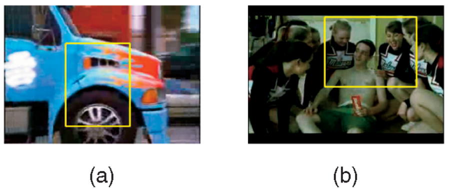


Fig. 11. Sample frames where the proposed method does not perform well. (a) Soda. (b) Candy.

version as follows: Frames with high scores are pseudo-labeled as positive, and the method is run again.

This method provides very good results when the OOI is highly textured, for example, in the cat food video (Fig. 7e). In some other videos, the performance is poor. We tried an alternative method where, instead of assigning a score based on the number of good matches, frames are classified as positive or negative based on nearest neighbor classification using both positive and negative frames. The performance is much worse than Method-E.

The results reported do not include the geometric verification step in [34] due to an average drop in performance from 62.3 to 61.1 percent. In only 3 out of 15 videos did the verification step improve performance. It should be noted that the verification step in [34] was based on the difference of Gaussian (DoG) function while we are using the MSER. The DoG is not affine invariant, while the MSER is. Besides, the MSER has the highest repeatability score among affine invariant region detectors as reported in [60]. An explicit geometric verification step could, therefore, provide more benefit to DoG than to MSER. In addition, the tracking process in our system can follow the gradual change of the video objects even when the verification step is not performed or when there is not much geometric consistency across different views.

5.1 Using Window-Level Labels

Throughout this paper, the discussion and experiments were based on frame-level labels. What if window-level annotation is available? In Methods-F, G, and H, we experiment with using window-level labels for classification.

Method-F. Method-F is an extension of Method-E, the matching-based method. When performing matching, we only use the SIFT descriptors that fall within the annotated window containing the object of interest, instead of using the whole image. We can see that the performance is better than that of Method-E.

Method-G. As in Method-F, whenever labeled windows are available, we extract features only within the window, but this time using Discrete AdaBoost as classifier. The performance is better than that of Method-A.

Method-H. To better understand why Method-C performs so well, we conduct a comparative experiment which tells us the "upper bound" performance of Method-C. Instead of sampling windows from the window-proposal distribution, we use the user-labeled windows in the following manner: We sample eight windows that are cocentered with the annotated window but with larger scales (each one increasing 10 pixels in width and height), and one additional window that covers the whole image. The classifier used is S-MILBoost. The performance is better than that of Method-C.

Summary. Experiments with Methods-F, G, and H show that when window-level annotation is available, the overall performance increases, at the expense of more labeling effort from the user's side. However, Method-C achieves nearly the same performance as Method-H, despite having only frame-level annotations. This shows the promise of a frame-level user labeling system and validates our idea of using multiple windows. Different windows contain different amount of background context, and Multiple Instance Learning automatically decides the right amount of background context required for classification.

6 Conclusion

We have presented an approach for removing irrelevant frames in a video by discovering the object of interest. Through extensive experiments, we have shown that this is not easily achieved by directly applying supervised or semisupervised learning methods in the literature developed for still images.

On a higher level, our method can be considered as a "weakly initialized" tracking system but without manual track initialization; The system finds out automatically what the "best track" is, with the objective of agreeing with the user's labeling on which frames contain the object of interest.

There are several future directions of interest. Currently, the spatial distribution in Section 4.2 is suitable for one or zero OOI in a frame. But this is not a problem for our system because, as long as one of the possibly many OOIs is discovered, the frame probability in Section 4.1 will be high. In other words, we don't need to identify every OOI in order to decide if a frame is relevant or irrelevant. If, in addition to determining the relevance of a frame, one also wants to discover all OOIs, one possible extension would be to use a mixture of Gaussians as spatial distribution. Having more parameters to estimate, however, is likely to decrease the performance, since the amount of user labels is very limited.

As we noted in the experiments, the performance depends directly on the placing of the windows. It would be helpful to adapt the window-proposal distribution's α parameter so that more windows are sampled from the Gaussian-near-uniform distribution when the estimated spatial distribution is bad. This might seem like a tautological statement since we don't know where the OOI is located. However, by means of estimating the quality of data fitting, this is potentially achievable. For example, if by examining the data log-likelihood we have a means of measuring the fitness of the spatial distribution, then we could adjust the value of α in (1) to control the number of windows being sampled from the estimated spatial distribution versus from the Gaussian-near-uniform distribution.

References

[1] D. Liu, G. Hua, and T. Chen, "Videocut: Removing Irrelevant Frames by Discovering the Object of Interest," *European Conf. Computer Vision,* vol. 1, pp. 441-453, 2008.

[2] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *Int'l J. Computer Vision,* vol. 56, pp. 151-177, 2004.

[3] P. Viola and M. Jones, "Robust Real-Time Face Detection," *Int'l J. Computer Vision,* vol. 57, pp. 137-154, 2004.

[4] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[5] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Object Level Grouping for Video Shots," *Int'l J. Computer Vision,* vol. 67, pp. 189-210, 2006.

[6] O. Maron and T. Lozano-Perez, "A Framework for Multiple Instance Learning," *Proc. Advances in Neural Information Processing Systems,* 1998.

[7] Y. Chen, J. Bi, and J. Wang, "MILES: Multiple Instance Learning via Embedded Instance Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 1931-1947, Dec. 2006.

[8] P. Viola, J. Platt, and C. Zhang, "Multiple Instance Boosting for Object Detection," *Proc. Advances in Neural Information Processing Systems,* 2005.

[9] M. Brand, N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 994-999, 1997.

[10] P. Smith, N. Lobo, and M. Shah, "Temporalboost for Event Recognition," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 733-740, 2005.

[11] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[12] R. Fergus, P. Perona, and A. Zisserman, "A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 380-387, 2005.

[13] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *Int'l J. Computer Vision*, vol. 61, no. 1, pp. 55-79, 2005.

[14] B. Leibe, K. Schindler, and L. Gool, "Robust Object Detection with Interleaved Categorization and Segmentation," *Int'l J. Computer Vision*, vol. 77, pp. 259-289, 2008.

[15] L. Fei-Fei, R. Fergus, and P. Perona, "One-Shot Learning of Object Categories," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594-611, Apr. 2006.

[16] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond Sliding Windows: Object Localization by Efficient Subwindow Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[17] K. Nigam and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-Training," *Proc. Int'l Conf. Information and Knowledge Management*, 2000.

[18] Y. Li, H. Li, C. Guan, and Z. Chin, "A Self-Training Semi-Supervised Support Vector Machine Algorithm and Its Applications in Brain Computer Interface," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, 2007.

[19] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," *Proc. IEEE Workshop Applications of Computer Vision*, 2005.

[20] I. Cohen, F. Cozman, N. Sebe, M. Cirelo, and T. Huang, "Semi-Supervised Learning of Classifiers: Theory, Algorithms and Their Applications to Human-Computer Interaction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 12, pp. 1553-1567, Dec. 2004.

[21] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering Objects and Their Location in Images," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.

[22] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and Their Extent in Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[23] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Learning Hierarchical Models of Scenes, Objects, and Parts," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1331-1338, 2005.

[24] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[25] J. Verbeek and B. Triggs, "Region Classification with Markov Field Aspect Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 17-22, 2007.

[26] L. Cao and L. Fei-Fei, "Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 1-8, 2007.

[27] D. Liu and T. Chen, "A Topic-Motion Model for Unsupervised Video Object Discovery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[28] D. Liu and T. Chen, "DISCOV: A Framework for Discovering Objects in Video," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 200-208, Feb. 2008.

[29] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, Inc., 1988.

[30] M. Jordan, *Learning in Graphical Models.* MIT Press, 1999.

[31] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning*, vol. 42, pp. 177-196, 2001.

[32] C. Schmid, "Weakly Supervised Learning of Visual Models and Its Application to Content-Based Retrieval," *Int'l J. Computer Vision*, vol. 56, pp. 7-16, 2004.

[33] O. Chum and A. Zisserman, "An Exemplar Model for Learning Object Classes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[34] D. Lowe, "Object Recognition from Local Scale Invariant Features," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1150-1157, 1999.

[35] X. Zhou and T. Huang, "Relevance Feedback in Image Retrieval: A Comprehensive Review," *ACM Multimedia Systems J.*, vol. 8, pp. 536-544, 2003.

[36] A. Torralba, "Contextual Priming for Object Detection," *Int'l J. Computer Vision*, vol. 53, pp. 169-191, 2003.

[37] A. Rabinovich, A. Vedaldi, C. Galleguillos, B. Wiewiora, and S. Belongie, "Objects in Context," *Proc. Int'l Conf. Computer Vision*, pp. 1-8, 2007.

[38] D. Parikh, L. Zitnick, and T. Chen, "From Appearance to Context-Based Recognition: Dense Labeling in Small Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[39] M. Riesenhuber and T. Poggio, "Hierarchical Models of Object Recognition in Cortex," *Nature Neuroscience*, vol. 2, pp. 1019-1025, 1999.

[40] M. Ranzato, F. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 1-8, 2007.

[41] B. Epshtein and S. Ullman, "Feature Hierarchies for Object Classification," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 220-227, 2005.

[42] Z. Tu, X. Chen, A.L. Yuille, and S.C. Zhu, "Image Parsing: Unifying Segmentation, Detection, and Recognition," *Int'l J. Computer Vision*, vol. 63, pp. 113-140, 2005.

[43] S. Fidler, G. Berginc, and A. Leonardis, "Hierarchical Statistical Learning of Generic Parts of Object Structure," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 182-189, 2006.

[44] B. Ommer and J.M. Buhmann, "Learning the Compositional Nature of Visual Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 1-8, 2007.

[45] S. Todorovic and N. Ahuja, "Unsupervised Category Modeling, Recognition, and Segmentation in Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2158-2174, Dec. 2008.

[46] F. Fleuret and D. Geman, "Coarse-to-Fine Face Detection," *Int'l J. Computer Vision*, vol. 41, pp. 85-107, 2001.

[47] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg, "Webcam Synopsis: Peeking Around the World," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.

[48] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proc. British Machine Vision Conf.*, 2002.

[49] http://www.robots.ox.ac.uk/~vgg/research/affine/, 2010.

[50] J. van de Weijer and C. Schmid, "Coloring Local Feature Extraction," *Proc. European Conf. Computer Vision*, 2006.

[51] B. Julesz, "Textons, the Elements of Texture Perception and Their Interactions," *Nature*, vol. 290, pp. 91-97, 1981.

[52] M. Isard, "Pampas: Real-Valued Graphical Models for Computer Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 613-620, 2003.

[53] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting Algorithms as Gradient Descent," *Proc. Advances in Neural Information Processing Systems*, 1999.

[54] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.*, vol. 39, pp. 1-38, 1977.

[55] B. Cestnik, "Estimating Probabilities: A Crucial Task in Machine Learning," *Proc. European Conf. Artificial Intelligence*, pp. 147-149, 1990.

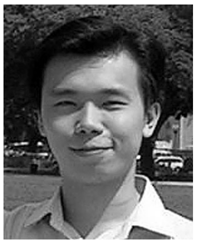
[56] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

[57] http://www-nlpir.nist.gov/projects/trecvid/, 2010.

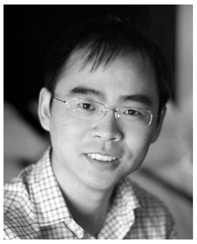
[58] Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences*, vol. 55, pp. 119-139, 1997.

[59] K. Bennett, A. Demiriz, and R. Maclin, "Exploiting Unlabeled Data in Ensemble Methods," *Proc. Int'l Conf. Knowledge Discovery and Data Mining*, 2002.

[60] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision*, vol. 65, pp. 43-72, 2005.



David Liu received the BS and MS degrees in electrical engineering from the National Taiwan University in 1999 and 2001, and the PhD degree in computer engineering from Carnegie Mellon University in 2008. He has been with Siemens Corporate Research, Princeton, New Jersey, since 2008. He was a research intern at Microsoft Live Labs Research, Bellevue, Washington, in 2007. He was a recipient of the Outstanding Teaching Assistant Award from the ECE Department at Carnegie Mellon University in 2006, the Taiwan Merit Scholarship since 2005, the best paper award from the Chinese Automatic Control Society in 2001, the Garmin Scholarship in 2001 and 2000, and the Philips Semiconductors Scholarship in 1999. He is a member of the IEEE.



Gang Hua was enrolled in the Special Class for the Gifted Young of Xian Jiaotong University (XJTU) in 1994 and received the BS degree in automatic control engineering from XJTU in 1999. He received the MS degree in control science and engineering at in 2002 from XJTU, and the PhD degree from the Department of Electrical and Computer Engineering, Northwestern University in 2006. He has been with the Nokia Research Center Hollywood since August 2009, where he is currently a researcher. From 2006 to 2009, he was a scientist at Microsoft Live Labs Research. His main research interests include computer vision, computer graphics, and machine learning. He received the Walter P. Murphy Fellowship from Northwestern University in 2002. When he was at XJTU, he was awarded the Guanghua Fellowship, the EastCom Research Scholarship, the Most Outstanding Student Exemplar Fellowship, the Sea-Star Fellowship, and the Jiangyue Fellowship in 2001, 2000, 1997, 1997, and 1995, respectively. He was also a recipient of the University Fellowship for Outstanding Student at XJTU from 1994 to 2002. He is a member of the IEEE.

Tsuhan Chen received the BS degree in electrical engineering from National Taiwan University in 1987. He received the MS and PhD degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1990 and 1993, respectively. He has been with the School of Electrical and Computer Engineering, Cornell University, Ithaca, New York, since January 2009, where he is a professor and a director. From October 1997 to December 2008, he was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, as a professor and an associate department head. From August 1993 to October 1997, he worked at AT&T Bell Laboratories, Holmdel, New Jersey. He served as the editor-in-chief for the *IEEE Transactions on Multimedia* in 2002-2004. He also served on the editorial board of the *IEEE Signal Processing Magazine* and as an associate editor for the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Signal Processing*, and the *IEEE Transactions on Multimedia*. He coedited a book titled *Multimedia Systems, Standards, and Networks*. He received the Charles Wilts Prize from the California Institute of Technology in 1993. He was a recipient of the US National Science Foundation CAREER Award, from 2000 to 2003. He received the Benjamin Richard Teare Teaching Award in 2006, and the Eta Kappa Nu Award for Outstanding Faculty Teaching in 2007. He was elected to the Board of Governors, IEEE Signal Processing Society, 2007-2009, and was a distinguished lecturer fo the IEEE Signal Processing Society, 2007-2008. He is a member of the Phi Tau Phi Scholastic Honor Society and a fellow of the IEEE.