A Generic Framework for Video Annotation via Semi-Supervised Learning

Tianzhu Zhang, *Member, IEEE*, Changsheng Xu, *Senior Member, IEEE*, Guangyu Zhu, Si Liu, and Hanqing Lu, *Senior Member, IEEE*

***Abstract*—Learning-based video annotation is essential for video analysis and understanding, and many various approaches have been proposed to avoid the intensive labor costs of purely manual annotation. However, there lacks a generic framework due to several difficulties, such as dependence of domain knowledge, insufficiency of training data, no precise localization and inefficacy for large-scale video dataset. In this paper, we propose a novel approach based on semi-supervised learning by means of information from the Internet for interesting event annotation in videos. Concretely, a *Fast Graph-based Semi-Supervised Multiple Instance Learning (FGSSMIL)* algorithm, which aims to simultaneously tackle these difficulties in a generic framework for various video domains (e.g., sports, news, and movies), is proposed to jointly explore small-scale expert labeled videos and large-scale unlabeled videos to train the models. The expert labeled videos are obtained from the analysis and alignment of well-structured video related text (e.g., movie scripts, web-casting text, close caption). The unlabeled data are obtained by querying related events from the video search engine (e.g., YouTube, Google) in order to give more distributive information for event modeling. Two critical issues of FGSSMIL are: 1) how to calculate the weight assignment for a graph construction, where the weight of an edge specifies the similarity between two data points. To tackle this problem, we propose a novel *Multiple Instance Learning Induced Similarity (MILIS)* measure by learning instance sensitive classifiers; 2) how to solve the algorithm efficiently for large-scale dataset through an optimization approach. To address this issue, Concave-Convex Procedure (CCCP) and nonnegative multiplicative updating rule are adopted. We perform the extensive experiments in three popular video domains: movies, sports, and news. The results compared with the state-of-the-arts are promising and demonstrate the effectiveness and efficiency of our proposed approach.**

***Index Terms*—Broadcast video, concave-convex procedure (CCCP), event detection, graph, Internet, multiple instance learning, semi-supervised learning, web-casting text.**

Manuscript received August 22, 2011; revised January 17, 2012 and March 07, 2012; accepted March 07, 2012. Date of publication April 03, 2012; date of current version July 13, 2012. This work was supported in part by 973 Program (Project No. 2010CB327905, 2012CB316304) and in part by the National Natural Science Foundation of China (Grant No. 60833006, 61070104, 90920303). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chia-Wen Lin.

T. Zhang is with the Advanced Digital Sciences Center (ADSC), Singapore 138632, and also with China-Singapore Institute of Digital Media, Singapore 119613.

C. Xu, S. Liu, and H. Lu are with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with China-Singapore Institute of Digital Media, Singapore 119613.

G. Zhu is with the Department of Electrical and Computer Engineering, National University of Singapore, and also with China-Singapore Institute of Digital Media, Singapore 119613.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2012.2191944

I. Introduction

With the exponential growth of social media in Web 2.0, the huge volume of videos being transmitted and searched on the Internet has increased tremendously. For example, in YouTube, over 48 h of new videos are uploaded to the site every minute, and more than 14 billion videos were viewed in May 2010 [1]. It is urgently required to make the unstructured multimedia data accessible and searchable with great ease and flexibility. Automatic video annotation is particularly crucial to understanding video semantic concepts for video summarization, indexing and retrieval purposes. Therefore, extensive research efforts have been devoted to event annotation for video analysis [2]–[5].

In this work, we attempt to detect event and annotate video based on its content in various video domains (e.g., sports, news, and movies). Our system is shown in Fig. 1, which has the following three features: 1) Event Recognition: Given a video, our system analyzes its content and obtains its event categories. In this way, each video can be labeled using their content information instead of its textual metadata. 2) Event Localization: For an event in a video sequence, our system is able to exactly locate its start/end boundaries. Therefore, interesting events in a video can be localized and video can be segmented with its semantic concept. 3) Semantic Search and Navigation: Based on event detection (event recognition and localization), our system can provide a service on video semantic navigation and search. As a result, it is very convenient for users to search and browse videos with their personal preferences.

There are many existing event detection approaches, which rely on text information or domain knowledge of the video, and employ labeled samples to train event models. However, the methods depend on the text information, such as the title, tags or surrounding page-text of the video, which ignore the richness of information within the video. Moreover, the domain knowledge is diverse in different video genres, which is limited and unsuitable to build a generic event model. The ambiguous video cues, background clutter and variant changes of camera motion, etc., further complicate the video analysis and impede the implementation of event detection systems. For general automatic video annotation methods, statistical models are first learned over labeled samples, then the annotation of unlabeled samples can be predicted with these models. However, this process has a major drawback that the labeled data are limited and, thus, the distribution of the labeled data cannot describe the distribution of the entire data set (including the labeled and unlabeled data) well. This kind of insufficiency of labeled data usually leads to inaccurate annotation results. Due to these factors, it is very difficult to design a generic algorithm to unify video annotation in

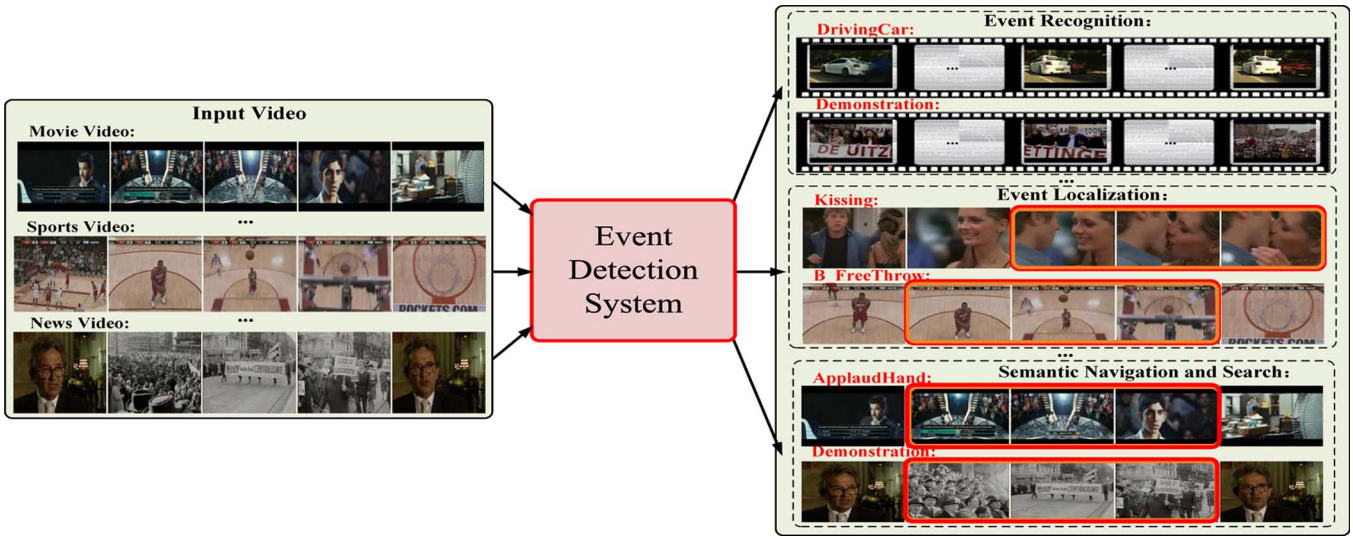


Fig. 1. Our proposed event detection system for different video domains. For better viewing, please see the color version online.

different domains (e.g., sports, news, and movies) with a high accuracy.

To tackle these issues, currently most of techniques for video annotation rely on video content and semi-supervised learning [6]. By leveraging unlabeled data with certain assumptions, semi-supervised learning methods are promising to deal with the insufficiency of training data just with a few labeled data. For small-scale labeled data, one can exploit the expert supervisory information in text source [4], [7] to reduce the human labor-intension, such as movie scripts, web-casting text and closed captions, which can provide useful information to locate possible events in video sequences. Because it is very cost-expensive and time-consuming to collect large-scale training data by text analysis, and there are still many videos without the corresponding text information for use, we attempt to make use of Internet to obtain large-scale collection of videos as unlabeled data to improve the performance of video annotation. As we know, Internet is a rich information source with many video events taken under various conditions and roughly annotated. By doing this, we adopt the semi-supervised learning algorithm to exploit the expert labeled and unlabeled video data together.

For our data, it is difficult to label the precise localization of interesting events for training model. The labeled data by text analysis only have the weakly associated labels, which means we know the video's label, but there may be no precise information about the localization of event in video. For the unlabeled data by Internet searching, we also do not know the precise localization of an event. To tackle this difficulty, we consider videos as multiple instance representation. That is, we temporally cut a video into multiple segments and find the segments corresponding to the event. Thus, event localization can be considered as a typical multiple instance learning problem, where each segment is an instance and all segments of a video clip compose a bag. By this formulation, video annotation can be obtained with the selection of more better segments.

Considering these problems, we propose a generic algorithm to automatically detect events from three realistic and challenging video datasets: sports, movies, and news. In this algorithm, we aggregate two sources of video data from text analysis and web video search together under a semi-supervised learning framework, and adopt multiple instance learning to gain the localization of event in video data. Therefore, we propose a Fast Graph-based Semi-Supervised Multiple Instance Learning (FGSSMIL) algorithm. Compared with the existing approaches, the contributions of our work can be summarized as follows.

1) We formulate video annotation as a multiple instance representation problem, which is most effective and suitable to localize the *interesting events* within the video.
2) We build a generic framework for video annotation in variant video genres by combination of multi-modality information (visual and audial feature) via the proposed FGSSMIL algorithm, which improves the detection performance and solves the insufficient training data problem by resorting to Internet data source.
3) To construct a discriminative graph for event model training, we propose a *Multiple Instance Learning Induced Similarity (MILIS)* measure, which considers the class structure ignored by the existing similarity measures [8], [9].
4) To solve FGSSMIL algorithm efficiently for large-scale data, we make use of concave-convex procedure and nonnegative multiplicative updating rule.

In our previous work [10], preliminary results of event detection on three datasets were reported. Compared with [10], a number of improvements have been made in this paper.

1) In [10], the instance discrimination is proved to be useful for de-noising and is not adopted for event modeling. In this paper, we use this information as a constraint for event model introduced in Section IV-B3. Experimental results demonstrate that embedding this information is efficient for video annotation.
2) The computational cost is extremely high for large-scale dataset in paper [10]. To tackle this problem, we use an efficient nonnegative multiplicative updating procedure to iteratively refine the solution and reduce the computational burden.

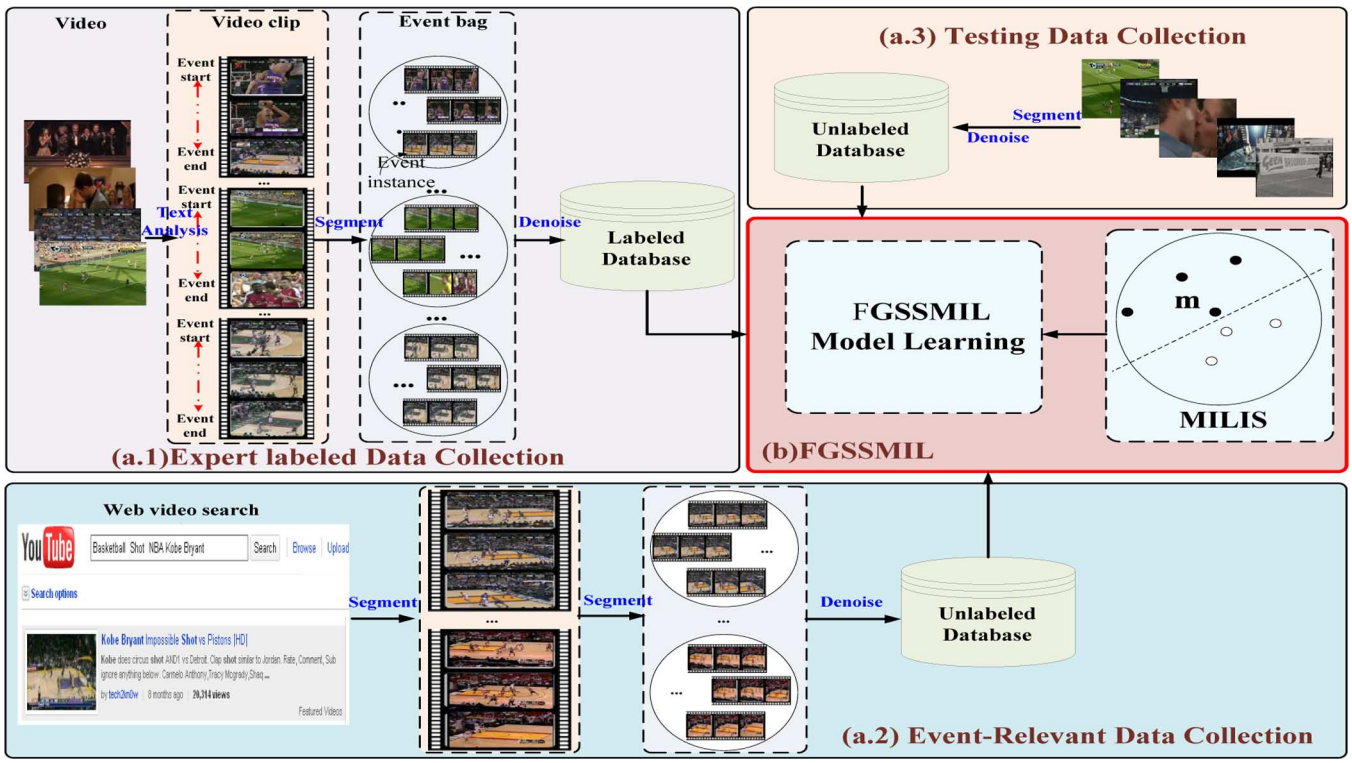


Fig. 2. Flowchart of the proposed video annotation algorithm. For better viewing, please see the color version online.

The flowchart of our proposed approach is illustrated in Fig. 2. It contains two primary parts: video collection, FGSSMIL. For video collection, it includes three kinds of data: expert labeled data collection, event-relevant data collection and testing data collection. For expert labeled data collection, text information of different kinds of video is used to structure the video segment and detect the start and end boundaries of the event to get a video event clip. Based on overlapped segmentation and de-noising of these video clips, a small-scale expert labeled database is collected. By querying keywords from the web, we obtain some raw videos. After segmentation and de-noising, a large-scale event-relevant database is constructed. To obtain testing video data, we collect them from the web, and the ground truth labels are obtained manually. In the process of de-noising, a Bayesian rule to efficiently remove some noise is adopted. For the FGSSMIL module, we combine three kinds of data for video annotation. To effectively obtain the similarity measure for the graph construction, we present an MILIS measure by considering the class structure. Finally, based on the learned event model, event recognition and localization are realized. The proposed approach is evaluated on highly challenging data from different video domains: movies, sports and news, and the experimental results are encouraging. The technical detail of each module will be described in the following sections.

The rest of the paper is organized as follows. Section II reviews the related work. The technical details of video collection and graph-based label propagation are presented in Sections III and IV, respectively. Experimental results are reported and analyzed in Section V. Finally, we conclude the paper with future work in Section VI.

II. Related Work

Extensive research efforts have been devoted to video analysis in recent years. The most related work to our method is video annotation via semi-supervised learning, event detection in different video domains, similarity measure for graph construction and multiple instance representation for video annotation. We review the state-of-the-arts of these four topics, respectively.

A. Video Annotation via Semi-Supervised Learning

Over the recent years, the availability of large data collections associated with only limited human annotation has turned the attention of a growing community of researchers to the topic of semi-supervised learning [6]. By leveraging unlabeled data based on certain assumptions, semi-supervised learning methods are expected to build more accurate models than those that can be achieved by purely supervised learning methods. Many different semi-supervised learning algorithms have been proposed. Some often-applied ones include self-training, co-training, transductive SVM, and graph-based methods. Extensive reviews of these methods can be found in [6]. Several of these methods have already been applied in video annotation and search [5], [11], [12].

In the past few years, the graph-based semi-supervised learning approach has attracted a lot of attention due to its elegant mathematical formulation and effectiveness in combining labeled and unlabeled data through label propagation [13]–[16]. This directly motivates our work in this paper. However, different from their approach, our proposed graph-based semi-supervised multiple instance learning method is very

suitable for video annotation. Moreover, our method learns optimal graph weights, and it is capable of obtaining good performance.

B. Event Detection in Different Video Domains

Event detection for various applications has been studied, and detailed surveys on this topic can be found in [17] and [18]. Most of existing work in event detection focuses on one type of video, such as movies, sports or news.

For event detection in movies, much work [19]–[21] incorporates visual information, closed-captioned text, and movie scripts to automatically annotate movies for classification and retrieval of videos. For event detection in sports video, most of the previous work is based on audio/visual/textual features directly extracted from video content [22]–[25]. These approaches heavily rely on audio/visual/textual features directly extracted from the video content itself. Some work uses text information [3], [2] such as close caption and web text for event analysis. Similar to the event detection in sports video, there is a lot of work for event analysis in news video [26]–[28]. By exploiting the available audio, visual and closed-caption cues, the semantically meaningful highlights in a news video are located and event boundaries are extracted.

For video surveillance, analysis and modeling of abnormal event detection has also been studied [29]–[31]. These methods can be broadly categorized according to the type of scene representation. One very popular category is based on trajectory modeling [29], [30], and the other is based on motion and appearance representations [31]. However, these approaches are unsuitable for multiple events detection in complex videos. Most of existing work uses domain knowledge [19], [3], which is difficult to be used for other video domains. For example, the methods used for event detection in movies cannot be applied to other domains such as sports videos that do not provide associated scripts; event detection approaches in sports video using text analysis [3] cannot be applied to many videos without text information.

Different from the precious work, we propose a generic framework for event detection in different video domains. The proposed method uses the videos with text information to learn model and then propagate labels to those videos with or without text information. Our learned model is based on video content. Therefore, it can be used for more generic video event analysis.

C. Similarity Measure for Graph Construction

More recently, graph-based methods have attracted the interest of researchers in this community due to their effectiveness and computational efficiency (most graph-based methods can be implemented with an efficient iterative process). A lot of work has demonstrated that the graph-based methods are computationally efficient with rather low computational costs. The weight of the edge is the core component of a graph, which is crucial to the performance of the semi-supervised learning.

The popular methods for the weight assignment include K-nearest neighbor (KNN), Gaussian kernel similarity (GKS) [8], and sparsity induced similarity measure (SIS) [9] based on sparse decomposition in L_1 norm sense. The main drawback of these approaches is that their performance is sensitive to the parameter variation and they do not take the label information into consideration. Different from the previous methods, we propose a new approach to measure the similarities based on class structure information.

D. Multiple Instance Representation for Video Annotation

There is little work [32] to detect events for video annotation with the multiple instance representations [33], [34], which is most suitable for event annotation in the videos. For multiple instance representations, each segment of a video clip is an instance and all segments of a video clip form a bag. Labels (or events) are attached to the bags while the labels of instances are hidden. The bag label is related to the hidden labels of the instances as follows: the bag is labeled as positive if any instance in it is positive, otherwise it is labeled as negative.

For our task, it is effective to distinguish a positive event instance (i.e., the particular event) from a negative instance (i.e., the background frames) with the help of multiple instance representation. In this way, our method can localize the interesting event boundary successfully.

III. Video Collection

Because there is little work to handle different kinds of video, no publicly generic dataset is available. In this section, we show how to collect video samples to construct three different video datasets (movie dataset, news dataset, and sports dataset). To avoid manually labeling a large amount of video data, we design a smart strategy to automatically collect videos from professional broadcast service providers and Internet. The small part of labeled videos are obtained by the analysis and alignment of well-structured video related text (e.g., movie scripts, web-casting text, close caption), while the large part of event-relevant videos and testing videos are collected from the Internet by querying related events from the video search engine (e.g., YouTube) and filtering noise. The details are as follows.

A. Expert Labeled Data by Text Analysis

Here, we introduce an automatic procedure, as shown in Fig. 1(a.1), for collecting videos from multiple video types (sports, movies, and news) supported by professional broadcast service providers. For different data sources, there are different available text information. For movie videos, we follow [19], [21], and [35] using scripts and subtitles to detect events to automatically collect training samples. For sports videos, we use web-casting text, which is usually available online and provided freely by almost all broadcasters [3], [36]. Keywords by which events are labeled are first predefined, then the time stamps where events happen are extracted from well-defined syntax structure syntax web-casting texts by using the keywords as input query key to a commercial software, dtSearch [37]. Similar to sports videos, the methods [26], [28] are adopted to find the temporal localizations of events with the closed-caption for news videos.

B. Event-Relevant Data From Internet

As we know, it is still very difficult to handle text analysis and time-alignment to collect enough labeled video data. Moreover, there are many videos without their corresponding text information. Therefore, we try to collect a large-scale event-relevant

data from the Internet to improve the performance of video annotation. We query the event labels on a web video search engine like YouTube or Google. Based on the assumption that the set of retrieved videos contains relevant videos of the queried event, we can construct a large-scale video dataset, which includes videos taken from multiple viewpoints in a range of environments. The challenge is how to use these videos, because content in the Internet is very diverse, which leads to the retrieved videos with much noise. For example, for a "Basketball Shot" query, a search engine is likely to retrieve some introduction videos of basketball shot. Our method must perform well in the presence of such noise. In this work, we adopt multiple keywords search ("Basketball Shot NBA Kobe Bryant") and propose an efficient method to remove some noise from the dataset in Section III-D. Compared with data obtained by text analysis, we call this collection as event-relevant dataset. In our semi-supervised learning algorithm, this dataset is used as unlabeled data.

C. Testing Data From Internet

We adopt the same way as event-relevant data collection from internet and obtain some testing video data. In our learning method, we put the testing data into our event model as unlabeled data, and propagate the label from labeled data to them. For testing, we obtain the ground truth event boundaries manually to evaluate the performance of our algorithm.

D. Segmentation and De-Noising by Bayesian Rule

To annotate video, we perform temporal segmentation of video clips and get segments composed of contiguous frames. Given a video clip v_i containing the event of interest but at unknown position within the clip, the clip v_i is represented by n_i temporally overlapping segments centered at frames $1, \ldots, n_i$ represented by histograms $h_i[1], \ldots, h_i[n_i]$. Each histogram captures the l_1-normalized frequency counts of quantized space-time interest points and audio features, as described in Section V-B. Let v_i^+ denote a positive video clip and v_i^- denote a negative video clip. v_{ij}^+ is the jth segment of a positive video clip v_i^+ and v_{ij}^- denotes the jth segment of a negative video clip v_i^-. Let $\{v_1^+, v_2^+, \ldots, v_m^+, v_1^-, v_2^-, \ldots v_n^-\}$ denote the set of m positive and n negative training video clips obtained by text analysis. $l(v_i) \in \{+1, -1\}$ is the bag label of v_i and $l(v_{ij}) \in \{+1, -1\}$ is the instance label of v_{ij}. For the negative video clips, all their segments are negative. However, for the positive video clips, their all segments must contain at least one true positive segment, and they may also contain many negative segments due to much noise, and imprecise localizations. The goal of de-noising is to identify the true positive segments in the positive video clips and remove some negative segments.

Given a true positive segment s, the probability that a segment v_{ij} is positive is calculated as follows:

$$\Pr(l(v_{ij}) = +1 \,|\, s) = \exp\left(-\frac{\|s - v_{ij}\|^2}{\delta_s^2}\right) \tag{1}$$

where $\|\bullet\|$ represents L2-norm, and δ_s is a parameter learned from the training data. Then, for a true positive segment s, the probability that a video clip v_i is a positive video clip is defined as follows:

$$\begin{aligned}\Pr(l(v_i) = +1 \,|\, s) &= \max_{v_{ij} \in v_i} \Pr(l(v_{ij}) = +1 \,|\, s) \\ &= \max_{v_{ij} \in v_i} \exp\left(-\frac{\|s - v_{ij}\|^2}{\delta_s^2}\right) \\ &= \exp\left(-\frac{d^2(s, v_i)}{\delta_s^2}\right)\end{aligned} \tag{2}$$

where $d(s, v_i) = \min_{v_{ij} \in v_i} \|s - v_{ij}\|$. In other words, the distance $d(s, v_i)$ between a segment s and all segments of a video clip v_i is simply equal to the distance between s and the nearest segment of v_i. Then $\Pr(l(v_i) = +1 \,|\, s) - \Pr(l(v_i) = -1 \,|\, s) = 2\exp(-(d^2(s, v_i))/(\delta_s^2)) - 1$. If $\Pr(l(v_i) = +1 \,|\, s) \geqslant \Pr(l(v_i) = -1 \,|\, s)$, we get $d(s, v_i) \leqslant \delta_s\sqrt{\ln 2}$. For a negative segment (i.e., false positive segment), however, its distances to the positive and negative video clips do not exhibit the same distribution as those from s. Therefore, given a true positive segment s, there exists a threshold θ_s which allows the decision function defined in (3) to label the video clips according to the Bayes decision rule:

$$h_{\theta_s}^s(v_i) = \begin{cases} +1 & \text{if } d(s, v_i) \leqslant \theta_s \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

where $\theta_s = \delta_s\sqrt{\ln 2}$ determined by training data as follows:

$$p_s = \max_{\theta_s} P_s(\theta_s) \tag{4}$$

where $P_s(\theta_s)$ is an empirical precision and defined as follows:

$$P_s(\theta_s) = \frac{1}{m+n} \sum_{i=1}^{m+n} \frac{1 + h_{\theta_s}^s(v_i) l(v_i)}{2}. \tag{5}$$

In this way, for each segment from the labeled dataset, we can obtain the p_s. Based on this value, we can remove some segments of each video clip. Note that the exact number of true positive segments for one specific positive video clip is unknown. To handle this problem, we propose that for a video clip: if $p_s > \mathrm{th}_1$, s is selected, where th_1 is a threshold and is manually set to be 0.5 in our experiments. Based on our experiments, this method is able to well solve our problem. In this way, we can remove irrelevant segments and video clips obtained by text analysis and construct an expert labeled dataset.

For the data obtained by web video search, we can also de-noise them by using the expert labeled data. Because data obtained from web have more noise than data obtained by text analysis, we set th_1 as 0.8 to obtain much cleaner data. Moreover, we adopt another strategy to confirm the reliability of the selected data from the web by the classifier introduced in Section IV-C. We can get segments for each video clip. Then, each segment s is classified using all classifiers trained with labeled data and has mean score Sc_s. If the score $Sc_s > \mathrm{th}_2$, this segment is selected and the video clip is viewed as event-relevant data. If scores of all segments of a video clip are below th_2, the video clip is not selected. In our experiments, the th_2

is manually set to be 0.7. After de-noising by the two strategies, we collect a large-scale of more cleaner event-relevant data from web, which will be used to give more distributive information.

IV. Fast Graph-Based Semi-Supervised Multiple Instance Learning (FGSSMIL) Algorithm

In this section, we introduce the FGSSMIL algorithm which combines expert labeled data, event-relevant data and testing data (as introduced in Section III) together to learn event model, and adopts multiple instance learning to detect the positive event instances from the event bags, where we consider the event with precise localization in a video clip as positive event instance and event with imprecise localization as negative event instance, and the corresponding video clip is viewed as an event bag. Next, we introduce the problem description in Section IV-A, and how to construct the event model and learn the similarity measure for the graph are presented in Sections IV-B1 and IV-C, respectively. Finally, we introduce how to solve the objective function in Section IV-D.

A. Problem Description

After video collection in Section III, each segment of a video clip is viewed as an instance (**event instance**), and all segments of the video clip comprise of a bag (**event bag**). To describe the problem simply, we just consider two classes, and it is easy to extend the formulation for multi-class problem introduced in Section IV-D. We use the following notation throughout this paper. Let $L = \{(x_1, y_1), \ldots, (x_{|L|}, y_{|L|})\}$ be the labeled data and let $U = \{x_{|L|+1}, \ldots, x_{|L|+|U|}\}$ be the unlabeled data. Each bag x_b is a set of instances $\{x_{b,1}, x_{b,2}, \ldots, x_{b,n_b}\}$, with its label denoted by $y_b \in \{0, +1\}$, where $+1$ is used for positive bag and 0 for negative bag. b is the index of bag, and n_b is the number of all instances of bag x_b. Each $x_{b,j} \in R^d$ is a d-dimensional feature vector representing an instance. Without loss of generality, we assume that the first L_1 bags are positive and the following L_2 bags are negative ($|L| = L_1 + L_2$). To describe the relationship between bags and instances, we use $x_{b,j} \in x_b$ to represent that $x_{b,j}$ is an instance from bag x_b and its label is represented as $y_{b,j}$.

Our task is to learn a soft label function $\hat{\boldsymbol{f}} : R^d \rightarrow [0, +1]$ that learns the label for each instance, we denote the predicted soft label of instance j by $f_j = \hat{\boldsymbol{f}}(x_{bj})$. Then, the labels of the bags can be calculated. We define a bag x_b's label f_b^* to be determined by the largest value of its corresponding instances' soft labels:

$$f_b^* = \max_{j, x_{b,j} \in x_b} f_j. \tag{6}$$

B. Event Model Formulation

1) Instance Similarity Constraint: In this section, we formulate the graph-based semi-supervised multiple instance learning in an instance-level way and define the cost criterion based on instance labels. Consider a graph $G = (V, E)$ with nodes corresponding to N feature vectors. There is an edge for every pair of the nodes. We assume that there is an $N \times N$ symmetric weight matrix $W = [w_{ij}]$ on the edges of the graph, where N is the number of all instances. The weight for each edge indicates the similarity between the two nodes that are connected by the edge. Intuitively, similar unlabeled samples should have similar labels. Thus, the label propagation can be formulated as minimizing the quadratic energy function [6]:

$$E_1(f) = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \tag{7}$$

where d_i is the sum of the ith row of W and denote $D = \mathrm{diag}(d_1, \ldots, d_N)$. f_i is the label of instance i, and f_i should be nonnegative. Assume the instance i is the ith instance of event bag b, f_i can be viewed as the probability for class c. We can obtain some prior knowledge from the labeled bag to its instance label, that is, f_i must be 0 if the bag x_b does not contain label c. In this way, bag label information is applied.

2) Instance Label Constraints by Bag Information: Equation (7) just controls the complexity in the intrinsic geometry of the data distribution and the smoothness of label over the instance-level graph. For our problem, we need to consider the constraints based on labeled bags. For a negative bag, it is straightforward to see that all instances in the bag are negative, i.e., $f_j = 0$, for all $j : x_{b,j} \in x_b$. Thus we have the penalty term:

$$E_2(f) = \sum_{b=1+L_1}^{|L|} \sum_{j: x_{b,j} \in x_b} f_j. \tag{8}$$

Meanwhile, for a positive bag, the case is more complex because a positive bag may contain negative instances as well. Actually, only one positive instance is necessary to determine a positive bag. Thus, we define the penalty term for a positive bag to be only related to the instance with the largest soft label:

$$E_3(f) = \sum_{b=1}^{L_1} \left(1 - \max_{j: x_{b,j} \in x_b} f_j \right). \tag{9}$$

3) Instance Discrimination Constraint: Instance discrimination is to describe the classification performance of a given sample, which reflects the discriminative power, and is proved to be useful for de-noising as shown in [10]. Considering the discriminative information, we make use of it to help our label propagation problem. The key point is how to obtain the instance discrimination. As we know, the positive instances in the class-specific event bags should have much discrimination to classify events labeled with the given class and other classes. Therefore, the discrimination of p_i can be calculated by its classification capability. For each instance, we can obtain its discrimination p_i as shown in (4).

For the instance discrimination information, we have the following observation: if the instance is positive, the value of p_i should be near to 1, and for the negative instance, the value is about 0. Based on the discrimination, we have the following cost function:

$$E_4(f) = \sum_i (f_i - p_i)^2. \tag{10}$$

$E_4(f)$ measures the data fitting capability, namely, the deviation between the estimated probabilistic confidence score f_i and the prior probability about discrimination p_i.

By combing the four items, we have the following cost criterion as our **event model**:

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2 + \alpha_1 \sum_{b=1+L_1}^{|L|} \sum_{j:x_{b,j}\in x_b} f_j + \alpha_2 \sum_{b=1}^{L_1}\left(1 - \max_{j:x_{b,j}\in x_b} f_j\right) + \alpha_3 \sum_i (f_i - p_i)^2 \quad (11)$$

where α_1, α_2 and α_3 are three parameters used to balance the weight. In our experiments, we set $\alpha_1 = \alpha_2 = 10, \alpha_3 = 5$, and obtained a good performance. Next, our aim is to minimize the objective function $E(f)$ in (11) to obtain the solution f.

Once we have found the optimal labels of the instances by minimizing the cost criterion $E(f)$, the bag-level label of any bag x_b can be calculated by taking the maximum value of its instances' labels using (6). The only two problems left are how to obtain efficient similarity measure $W = [w_{ij}]$ and how to solve the optimization task in (11). For the first problem, we propose a multiple instance learning based method to learn the similarity measure $W = [w_{ij}]$ and introduce this in Section IV-C. Due to the existence of the $\max(\bullet)$ function in the loss function (11) for positive bags, $E(f)$ is generally non-convex, and cannot be directly optimized. In Section IV-D, we will derive a sub-optimum solution to this problem.

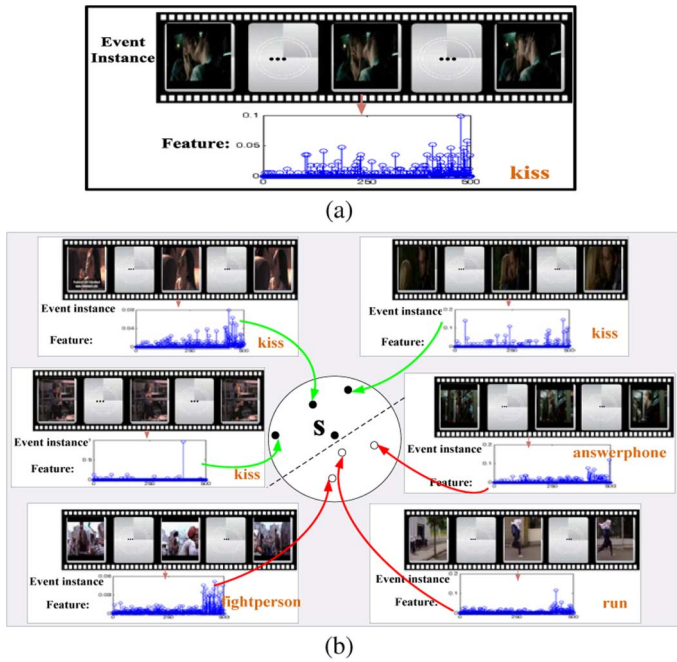


Fig. 3. Learning multiple instance learning induced similarity (MILIS) measure for each event instance s. (a) Event instance s, which belongs to one instance of kissing event bag. (b) Learning a classifier to describe the similarities of other event instances to s. "$\bullet$" represents similar event instance from the same class of event with s, and "o" represents unrelated event instance.

C. Multiple Instance Learning Induced Similarity (MILIS) Measure

There is one main drawback of most existing similarity measures, such as the Euclidean distance and Gaussian Kernel Similarity measure, which is that the similarity measurement completely ignores the class structure. For example, in Fig. 3, given an event instance s belonging to category c, it is possible that some event instances from the same class to s are less similar than the ones from other classes when a predefined and heuristic distance metric is adopted. To tackle this problem, we attempt a discriminative solution to get truly similarity by learning some classifiers. Here, we formulate the similarity measure learning as a problem of multiple instance learning (MIL) [38] and mi-SVM [39] is employed to solve the problem.

Next, we will introduce how to train a classifier for an event instance s from the category c. This training process can be repeated for all classifiers of different kinds of event instances. For the event instance s denoted as feature vector I_s, its classifier is trained in the hyper-sphere centered at I_s with radius of r_s in the feature space (as shown in Fig. 3). The training samples are the samples in class c denoted as positive bags and those in other categories denoted as negative ones. This strategy filters out the instances that are very different from s for each bag and enables the classifier to be learned only in the local feature space. Therefore, it is very efficient to reduce the computational burden and learn a discriminative classifier.

Define the distance from event instance I_s to event bag x_b as $d_{b,j,s} = \min_j \|I_s - x_{b,j}\|$, where $\|\bullet\|$ represents L2-norm. In practice, it is found quite robust and in majority cases the positive instance in the positive bag x_b is $\{x_{b,j^*} \mid j^* = \arg\min_j \|x_{b,j} - I_s\|\}$. Based on this observation, r_s is set as follows: $r_s = \text{mean}_{b\in\text{pos}}(d_{b,j,s}) + \beta \times \text{std}_{b\in\text{pos}}(d_{b,j,s})$, where β is a trade-off between efficiency and accuracy. In our experiments, the β is manually set to be 2.5 by experience. To be more efficient in experiments, within the hyper-sphere, at most k_p nearest event instances to I_s are selected for each positive event bag, and k_n for each negative event bag. $k_p = 5$ and $k_n = 2$ are used by experience.

Denote $y_{b,j}$ to be the instance label of event instance $x_{b,j}$ and y_b the label of event bag x_b, where $x_{b,j}$ is the feature of the event instance j in the event bag x_b. mi-SVM is formulated as follows:

$$\begin{aligned}
&\min_{\{y_{b,j}\}} \min_{w^*,b_0,\xi} \frac{1}{2}\|w^*\|^2 + C\sum_{x_{b,j}} \xi_{x_{b,j}} \\
&\text{s.t.} \sum_j \frac{y_{b,j}+1}{2} \geqslant 1, \quad \forall x_b \text{ s.t. } y_b = 1 \\
&y_{b,j} = -1, \ \forall x_b \text{s.t.} y_b = -1 \\
&\forall j : y_{b,j}(\langle w^*, x_{b,j}\rangle + b_0) \geqslant 1 - \xi_{x_{b,j}}, \xi_{x_{b,j}} \geqslant 0, \\
&y_{b,j} \in \{-1, 1\}.
\end{aligned} \quad (12)$$

Denote $\text{mi}-\text{SVM}_s$ as the trained classifier corresponding to event instance s. Based on this classifier, all event instances can be projected to real value with a function. For simplicity, the project function is defined as follows:

$$g_s(x_{b,j}) = \begin{cases} \text{mi}-\text{SVM}_s(x_{b,j}) \exists x_{b,j}, \text{ s.t. } \|x_{b,j} - I_s\| \leqslant r_s \\ 0 \quad \text{otherwise} \end{cases} \quad (13)$$

where $\text{mi}-\text{SVM}_s(x_{b,j}) \in \mathbb{R}$ is the output of the classifier $\text{mi}-\text{SVM}_s$ with the input $x_{b,j}$. Based on this score, the sim-

ilarity between instance s and $x_{b,j}$ can be simply defined as follows:

$$w_{sj} = \begin{cases} g_s(x_{b,j}) & \text{if } g_s(x_{b,j}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Therefore, for each event instance s in labeled data set, we can get its corresponding classifier and the similarities with other event instances. Though there may be some instances in positive event bags belonging to negative instances after de-noising. It is still efficient to learn the similarity measure based on our experimental results.

Based on the L labeled data and U unlabeled data, the similarity measure W can be splitted into labeled and unlabeled sub-matrices: $W = \begin{pmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{pmatrix}$, where $W_{LU} = W_{UL}$. W_{LL} and W_{LU} can be obtained by (14) using learned classifiers. For the unlabeled data, we adopt Euclidean distance to measure the similarity W_{UU} between data points.

D. Iterative Solution Using CCCP

Because $E_3(f)$ defined by (9) is non-convex, $E(f)$ can be viewed as that a convex function adds a concave function. Therefore, we adopt the constrained concave convex procedure (CCCP) to find the sub-optimum solution. CCCP is proposed in [40] as an extension of [41], and is theoretically guaranteed to converge. It works in an iterative way: at each iteration, the first order Taylor expansion is used to approximate the non-convex functions, and the problem is thus approximated by a convex optimization problem. The sub-optimum solution is given by iteratively optimizing the convex subproblem until convergence.

Note that $\max(\bullet)$ is not differentiable at all points. To use CCCP, we have to replace the gradients by the subgradients. For multi-class problem, let $l = [f^c_{b1}, \ldots, f^c_{bj}, \ldots, f^c_{bn_b}]^T$, where f^c_{bj} denotes the probability that the jth instance in the bth bag belongs to the cth class and $c \in \{1 \ldots C\}$, C is the total number of classes and n_b is the number of instances in the bth bag. We pick the subgradient with ρ, which is an $n_b \times 1$ vector and its jth element is given by

$$\rho_j = \begin{cases} \frac{1}{\tau} & \text{if } l_j^{(t)} = \max\left(l_b{}^{(t)}\right) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $\max(l_b^{(t)})$ represents the largest label value of bag b and τ is the number of instances with the label value $\max(l_b^{(t)})$. At the $(t+1)$th iteration, we estimate the current l based on $l^{(t)}$ and the corresponding ρ_j. As $\rho^T l^{(t)} = \sum_j \rho_j l_j^{(t)} = \max l^{(t)} \sum_{\rho_j \neq 0} \rho_j = \max l^{(t)}$, for the function $\max(l)$, its first order Taylor expansion is approximated as $(\max l)_{l^{(t)}} \approx \rho^T l$.

For the tth iteration of CCCP, the objective function in (11) is rewritten in matrix form as follows:

$$\begin{aligned} \min_F Q(F) = {} & \operatorname{Tr}(F^T L F) + \alpha_1 \sum_b \sum_c (1 - Y_{bc}) h_c F^T q_b \\ & + \alpha_2 \sum_b \sum_c Y_{bc} \left(1 - h_c \beta U_b F h_c^T\right) \\ & + \alpha_3 \operatorname{Tr}((F-P)^T (F-P)) \\ & \text{s.t. } F \geqslant 0, \quad F e_1 = e_2 \end{aligned} \tag{16}$$

where L is a Laplace matrix $L = D - W$, with D being the degree matrix and $\operatorname{Tr}(\cdot)$ represents matrix *trace* operator, $F = [f_{11}, \ldots, f_{1n_1}, \ldots, f_{Bn_1}, \ldots, f_{Bn_B}]^T$, $f_{bn_b} = [f^1_{bn_b}, \ldots f^c_{bn_b}, \ldots f^C_{bn_b}]^T$ and $F \in R^{N \times C}$. $P = [p_{11}, \ldots p_{1n_1}, \ldots, p_{Bn_1}, \ldots, p_{Bn_B}]^T$, and $p_{bn_b} = [p^1_{bn_b}, \ldots, p^c_{bn_b}, \ldots, p^C_{bn_b}]^T$ is the discrimination of instance n_b of bag b. B is the number of all bags and $N = \sum_{b=1}^{B} n_b$ is the number of all instances. Each row of F corresponds to the posterior probability distribution of an instance, hence should be: 1) positive, 2) l_1 normalized. Therefore, the constraint of (16) is necessary. $Y = [Y_{bc}]$, $Y_{bc} = 1$ if bag b belongs to the cth class, otherwise $Y_{bc} = 0$. h_c is a $1 \times C$ indicator vector, the cth element of which is one and others are zero, and $q_b = [\underbrace{0, \ldots, 0}_{1,\ldots,b-1}, \underbrace{1, \ldots, 1}_{b}, \underbrace{0, \ldots, 0}_{b+1,\ldots,B}]^T$ is an $N \times 1$ vector whose all elements, except for those elements corresponding to the bth bag, are zeros. β is a $C \times N$ matrix, $\beta = [\beta_1, \ldots, \beta_b, \ldots, \beta_B]$, each $\beta_b = [\beta^T_{b1}, \ldots, \beta^T_{bc}, \ldots, \beta^T_{bC}]^T$ is a $C \times n_b$ matrix corresponding to bag b and $\beta_{bc} = \eta^T$. $e_1 = \mathbf{1}_{C \times 1}$ and $e_2 = \mathbf{1}_{N \times 1}$ are both all-one vectors, α_1 and α_2 are two parameters used to balance the weight. $U_b = \operatorname{diag}(u_1, \ldots, u_b, \ldots u_B)$ is an $N \times N$ diagonal block matrix, where $u_k = 0_{n_k \times n_k}$ for $k = 1, \ldots, b-1, b+1, \ldots B$ and $u_b = I_{n_b \times n_b}$, I represents an identity matrix.

The subproblem in (16) is a standard quadratic programming (QP) [42] problem and can be solved by any state-of-the-art QP solvers. In the work [10], it is solved with global optimum using existing convex optimization packages, such as Mosek [43]. However, the computational cost is extremely high for large-scale dataset. Therefore, we adopt an efficient nonnegative multiplicative updating procedure to iteratively refine the solution and reduce the computational burden.

By relaxing the hard constraints of $Fe_1 = e_2$ into soft ones, we obtain the soft-version objective formulation:

$$\min_F Q(F) + \lambda \|Fe_1 - e_2\|^2, \text{ s.t. } F \geqslant 0 \tag{17}$$

where λ is a tunable weighting parameter and is manually set to be 35 in experiments. Then we get the following corollary on the solution to (17). The proof can be easily derived by calculating the derivative of the objective function in (17) with respect to F. The (17) can be optimized based on nonnegative multiplicative updating rule as shown in (18) at the bottom of the page.

$$F_{ij} \leftarrow F_{ij} \times \frac{\left[2WF + 2\lambda e_2 e_1^T + \alpha_2 \sum_b \sum_c F_{bc} U_b^T \beta^T h_c^T h_c + 2\alpha_3 F\right]_{ij}}{\left[2DF + 2\lambda F e_1 e_1^T + + \alpha_1 \sum_b \sum_c (1 - Y_{bc}) q_b h_c + 2\alpha_3 P\right]_{ij}} \tag{18}$$

Fig. 4. Event exemplar frames from the three different video types: movie video, sports video, and news video. There are 6 different types of events on movie dataset, 6 different types of events on sports dataset, and 4 different types of events on news dataset.

Running CCCP iteratively until convergence, we can obtain the sub-optimum solution for the instance labels. The label for each bag is then calculated as the largest label of all its instances using (6).

Out-of-Sample Extension: For a new testing instance t, its label is given as

$$f_t = \sqrt{d_t}\sum_j w(j,t)\frac{f_j}{\sqrt{d_j}} \Big/ \sum_j w(j,t) \tag{19}$$

where $w(j,t)$ represents the similarity between instance j and t. d_j and d_t have the same meaning as in (7). d_t is an unknown constant for a particular testing instance t, and we can ignore it when making decision. f_j is the obtained instance label by (11).

When given a new testing bag with multiple instances, we make use of (19) to obtain instance label and then apply (6) to classify the bag. Because we can obtain a label for each instance of a bag, the localization of event can be also obtained. Here, Gaussian kernel based temporal filtering is conducted to smooth the event instances from a video stream taking account of the temporal consistency of events.

V. Experimental Results

In this section, we present extensive experimental results on movies, sports and news video datasets in order to validate the proposed approach.

A. Datasets

As introduced in Section III, we obtain three different video datasets. For movie videos, we select 6 different kinds of representative events: AnswerPhone, DriveCar, Kissing, FightPerson, Run, and Applauding. We obtain from the web 981 unlabeled data, and 311 expert labeled data are obtained by text-video alignment. The dataset is from about 60 h of videos. For sports videos, we select 6 different kinds of events: Basketball Foul (B_Foul), Basketball Free Throw (B_FreeThrow), Basketball Run (B_Run), Basketball Shot (B_Shot), Soccer Foul (S_Foul), and Soccer Shoot (S_Shoot). We obtain from the web 913 unlabeled data, and 316 expert labeled data are obtained by text-video alignment. There are about 25 h of videos for this dataset. For news videos, we select 4 kinds of events: Eating, PlayingInstrument, Demonstration, and Dancing. We obtain from the web 863 unlabeled data, and 311 expert labeled data are collected by text-video alignment from about 20 h of videos. The number of test data is 253, 211, and 203 for the three datasets, respectively. Note that the three different kinds of videos are very challenging for video analysis due to its loose and dynamic structure as shown in Fig. 4.

B. Video Feature Extraction

Visual features and audio features are complimentary and important for video event detection. For example, Basketball Free Throw and Basketball Shot are most similar just using visual features, however, audio features, such as whistle of referee, are discriminative. On the contrary, visual features are very important to distinguish dancing and playing a guitar, because they both have similar background music and different motion features. In the following subsections, we will introduce two features, respectively.

1) Spatio-Temporal Features: Sparse space-time features have recently shown good performance for video analysis [44]. They provide a compact video representation and tolerance to background clutter, occlusions and scale changes. We detect interest points using a space-time extension of the Harris operator. To characterize motion and appearance of local features, we compute histogram descriptors of space-time volumes in the neighborhood of detected points. For a 3-D video patch in the neighborhood of each detected space-time interest point (STIP), it is partitioned into a grid with $3 \times 3 \times 2$ spatio-temporal blocks; 4-bin HOG descriptor and 5-bin HOF descriptor are then computed for all blocks and are concatenated into a 72-element and 90-element descriptors, respectively. The details can be found in [19]. For each volume we compute coarse histograms of oriented gradient (HoG) and optical flow (HoF). Normalized histograms are concatenated into HoG and HoF descriptor vectors and are similar in spirit to the well known SIFT descriptor. HoF is based on local histograms of optical flow. It describes the motion in a local region. HoG is a 3-D histogram of 2-D (spatial) gradient orientations. It describes the static appearance over space and time. Then the two descriptors are concatenated into one 162-dimensional vector, which is reduced by PCA to 60.

2) Audio Features: The mel-frequency cepstral coefficients (MFCCs) [45] are proved more efficient [46] for audio recognition. Therefore, we adopt the MFCCs to represent audio. The MFCCs are based on a short-term spectrum, where Fourier basis audio signals are decomposed into a superposition of a finite number of sinusoids. The power spectrum bins are grouped and smoothed according to the perceptually motivated

TABLE I
COMPARED RESULTS ABOUT THE AVERAGE ACCURACY AND COMPUTATIONAL COST OF GSSMIL [10] WITH AND WITHOUT DE-NOISING BY BAYESIAN RULE. THE m REPRESENTS MINUTES

Method / Dataset	With De-noising		Without De-noising	
	Accuracy	Computational Cost	Accuracy	Computational Cost
Movie	**51.27%**	**241m**	35.13%	387m
News	**63.80%**	**115m**	45.28%	179m
Sports	**57.71%**	**213m**	37.36%	313m

TABLE II
COMPARED RESULTS ABOUT THE AVERAGE ACCURACY AND COMPUTATIONAL COST BETWEEN OUR PROPOSED FGSSMIL AND GSSMIL [10]. THE m REPRESENTS MINUTES

Method / Dataset	FGSSMIL		GSSMIL	
	With p_i	Computational Cost	Without p_i	Computational Cost
Movie	**54.22%**	**23m**	51.27%	241m
News	**65.34%**	**9m**	63.80%	115m
Sports	**59.76%**	**18m**	57.71%	213m

Mel-frequency scaling. Then the spectrum is segmented by means of a filter bank that typically consists of overlapping triangular filters. Finally, a discrete cosine transform applied to the logarithm of the filter bank outputs results in vectors of decorrelated MFCC features. In our experiments, we use 13-dimensional MFCC features.

3) Bag of Features: For the two different modality features, we build bag-of-features (BoF), respectively. This requires the construction of visual vocabulary. In our experiments we cluster a subset of $400k$ features sampled from the training videos with the k-means algorithm for visual features. The number of clusters is set to $k = 400$ for visual features and $k = 100$ for audio features with a subset of $100k$ features sampled from the training data, which have shown empirically to give good results. The BoF representation then assigns each feature to the closest (we use Euclidean distance) vocabulary word and computes the histogram of visual word occurrences over a space-time volume corresponding to the segments obtained by over segmentation of the entire video clip. Then, for each segment, the two histograms are concatenated into one 500-dimensional vector and then normalized.

C. Recognition Evaluation About De-Noising

In this subsection, we evaluate the importance of de-noising using Bayesian rule. To obtain event boundaries, we perform temporal segmentation of video clips and get segments composed of contiguous frames. Finally, we select the much better segments to localize the event in each video clip. For a event relevant video clip, their all segments must contain at least one true positive segment, and they may also contain many negative segments due to much noise, and imprecise localizations. If we use all the segments of each video clip to construct the instance-level graph, the number of nodes is very large. Moreover, the noisy segments maybe have a bad effect on the similarity measure of graph. Therefore, it is necessary to adopt de-noising to remove some negative segments and improve the video annotation performance.

We run the GSSMIL algorithm [10], and compare the methods with de-noising and without de-noising. The results with de-noising and without de-noising are illustrated in Table I, which shows the average accuracy and computational cost on the three datasets, respectively. From the results, we can see the method using de-noising preprocess does not only improve the recognition performance, but also reduce the computational time due to removing many negative segments. Therefore, it is necessary and useful to adopt de-noising method to improve performance.

D. Recognition Evaluation About Instance Discrimination p_i

In this subsection, we evaluate the importance of instance discrimination for video annotation. For each instance, we can get its discrimination p_i as shown in (4), which takes into account its classification capability. Therefore, the instance discrimination p_i is class-specific and has the discrimination to distinguish events labeled with the given class and other classes. When the labels (f_i) are propagated from the labeled samples to unlabeled samples, it is necessary to consider the instance discrimination p_i as shown in (10) to build the graph model. In this way, our graph model has a constraint that the estimated probabilistic confidence score f_i should be consistent with its classification capability p_i. As a result, our event model as shown in (11) is much more discriminative than the model in [10].

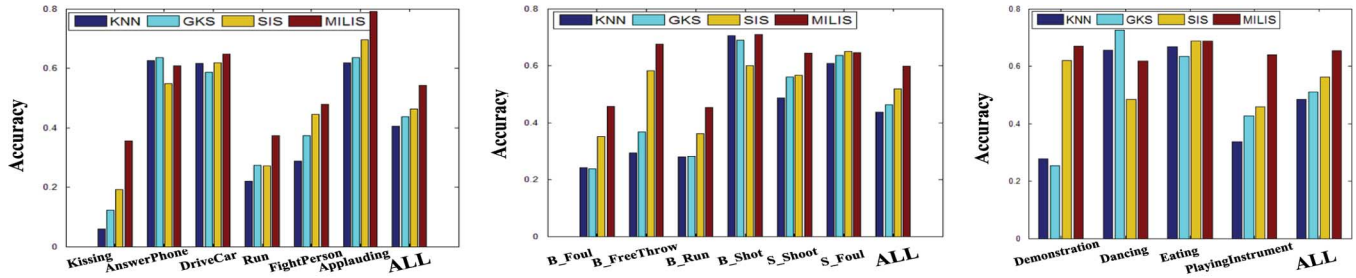
The compared results between our method FGSSMIL with p_i and GSSMIL [10] without p_i are given in Table II, which shows the average accuracies become better on the three dataset, and has about 2%–3% improvement. Based on the results, we can confirm that the instance discrimination p_i is efficient and discriminative for the probabilistic confidence score estimation f_i. Finally, it can improve the video event localization performance.

E. Recognition Evaluation About Computational Cost

In this subsection, we compare the computational cost between our method FGSSMIL and GSSMIL [10]. In [10], the event model is viewed as a standard QP problem and is solved with global optimum using existing convex optimization packages, such as Mosek [43]. However, the computational cost is extremely high for large-scale dataset. To make our proposed method be suitable for large-scale video annotation problem, we adopt an efficient nonnegative multiplicative updating procedure to iteratively refine the solution and reduce the computational burden.

The results about computational cost are shown in Table II on the three datasets, respectively. All the experiments are run on a sever with Intel Xeon(R) CPU with 2.4 GHz CPU and 16 GB memory and the code is run in MATLAB platform.

As shown in (7), our model is an instance-level graph, and the computational cost is determined by the number of instances. With the increase of instances, the computational cost will increase dramatically. However, CCCP converges quite fast and the nonnegative multiplication updating procedure only involves matrix multiplication, which cause our method FGSSMIL obtains the solution very fast. The complexity in (18) depends on the size of matrixes $W \in R^{N\times N}$ and $F \in R^{N\times C}$, where N is the number of instances, and C is the number of all classes. The running time for multiplying rectangular matrices (one $N \times N$-matrix with one $N \times C$-matrix)



(a) (b) (c)

Fig. 5. Experimental results about accuracy of each event class with different similarity measures (KNN, GKS, SIS, and MILIS) on the three datasets, respectively. (a) Movie Video Dataset. (b) Sport Video Dataset. (c) News Video Dataset.

is $O(N^2C)$, however, more efficient algorithms exist for fast matrix multiplication [47]. When updating F with the efficient nonnegative multiplicative updating procedure, the computational cost is $O(NC)$. Table II gives a quantitative comparison of computational cost between FGSSMIL and GSSMIL [10] in detail. For example, our method just requires 23 min, which is much faster than GSSMIL with more than 4 h on Movie dataset with 9816 instances. Therefore, our proposed algorithm is much more efficient than GSSMIL for video annotation, and can be applied into other large-scale problems.

F. Comparison of Different Similarity Measurements

In this subsection, we evaluate the performance of event recognition with different similarity measurements. We compare the proposed MILIS measure with three very popular and extensively used similarity measures: GKS, SIS [9], and KNN on the three datasets, respectively.

For GKS, we use $d_{ij} = \exp(-(\|p_i - p_j\|^2)/(\delta^2))$ to measure similarity and the variance δ is set to be 1.5, 1.5, 1.2 which achieved the best performance for the three datasets, respectively. For KNN, we use inner product similarity to find the K nearest neighbors while the number of nearest neighbors K is tuned by cross-validation. We found that 30, 30, 20 work better for our experiments on the three datasets, respectively. Then, the similarity values between a sample and its K nearest neighbors are their correlation coefficients while those between the sample and the rest are set to 0. As for SIS, we normalize all feature vectors so that their L_2 norms are 1 before computing the weight matrix.

Fig. 5 shows the propagation accuracies of four different similarity measures: KNN, GKS, SIS, and MILIS. The x-axis is categories of different events. The y-axis is label propagation accuracy for individual classes and the mean accuracy for all of the categories denoted as "ALL". We can see that MILIS (labeled as "MILIS") works the best. For other similarity measures, SIS (labeled as "SIS") works better than GKS, and GKS (labeled as "GKS") works better than KNN (labeled as "KNN").

Fig. 5 also shows that there are some classes where our approach does not outperform other approaches, such as AnswerPhone, Soccer Foul (S_Foul), and Dancing. To explain the reason, we give an example on news video dataset. We can see that two similarity measures (GKS and KNN) outperform our method (MILIS) for "Dancing" recognition. This is because the four classes are very prone to be classified as "Dancing" using GKS and KNN with our video features, which leads to a high performance for "Dancing" recognition with a high false alarm rate. In addition, we notice that the mean accuracies for all of concepts of our method outperform the other methods on the three datasets, respectively. The average precision values of different similarity measures on the three datasets are shown in Table III. We can see that our method has an improvement of about 8%.

Based on the mean accuracies for all of the concepts on the three datasets shown in Table III, we can confirm that our approach obtains the best performance compared among the existing methods. The result is obvious, because the existing similarity measures such as KNN, SIS and GKS completely ignore the class structure. However, our approach considers the distribution of feature points in the feature space and adopts a classifier to improve the similarity measure by using discriminative class information.

TABLE III
EXPERIMENTAL RESULTS ABOUT THE AVERAGE ACCURACY OF DIFFERENT SIMILARITY MEASURES (KNN, GKS, SIS, AND MILIS) ON THE THREE DATASETS, RESPECTIVELY

Method / Dataset	KNN	GKS	SIS	MILIS
Movie	40.45%	43.74%	46.18%	**54.22%**
News	48.52%	51.03%	56.27%	**65.34%**
Sports	43.60%	46.20%	51.81%	**59.76%**

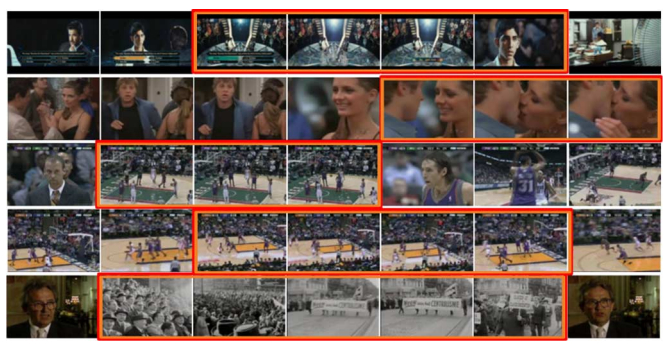
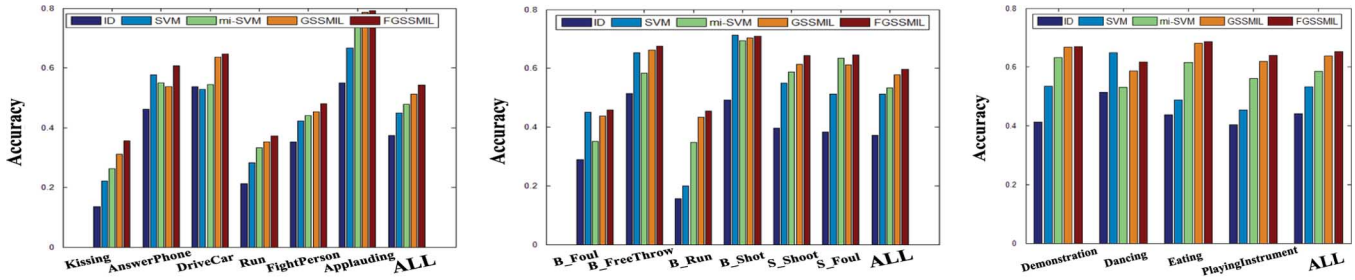


Fig. 6. Some examples of temporal localizations of events by the proposed algorithm on the three datasets. Each row shows example frames from the entire video clip. Example frames of automatically localized events within the clips are shown in red. The four rows represent "Kissing" and "Applauding" on movie video dataset, "B_FreeThrow" and "B_Shot" on sports video dataset, and "Demonstration" on news video dataset, respectively.



(a) (b) (c)

Fig. 7. Experimental results of different learning methods. The result with instance discrimination is denoted as "ID"; the result of an entire video as a training sample without segmentation shown as "SVM"; the result of just using labeled training data without unlabeled data is denoted as "mi-SVM"; the result in paper [10] is shown as "GSSMIL"; the result of our method is labeled as "FGSSMIL". (a) Movie Video Dataset. (b) Sports Video Dataset. (c) News Video Dataset.

TABLE IV
COMPARED RESULTS OF DIFFERENT LEARNING STRATEGY ON THE THREE DATASETS

Method / Dataset	ID	SVM	mi-SVM	GSSMIL	FGSSMIL
Movie	37.46%	44.99%	47.85%	51.27%	**54.22%**
News	44.17%	53.23%	58.51%	63.87%	**65.34%**
Sports	37.18%	51.30%	53.31%	57.71%	**59.76%**

Compared with [10], the average accuracies of four different similarity measures (KNN, GKS, SIS, and MILIS) are improved. The reason is that we adopt an efficient and effective nonnegative multiplicative updating rule to obtain perfect solution and mine some useful information from the instance discrimination for event modeling.

G. Comparison of Different Learning Methods

In this section, we try to demonstrate the effectiveness of our method by using of semi-supervised learning and multiple instance learning. Therefore, four experiments with different learning methods are performed. First, we just use the instance discrimination of testing data, and use this information as the label. The results are shown in Fig. 7 (denoted as "ID"). Second, we do not formulate the event detection into a multiple instance learning problem, instead, we use the entire video as a training sample and train an SVM classifier. The results are shown in Fig. 7 (labeled as "SVM"). In the third experiment, we do not use the unlabeled data from the Internet, and only employ the expert labeled data, its results are shown in Fig. 7 (labeled as "mi-SVM"). The last experiment is our learning method which formulates the video annotation as multiple instance learning problem and combines labeled and unlabeled data under a semi-supervised framework. The results in paper [10] are denoted as "GSSMIL", and our method is indicated as "FGSSMIL".

The average precisions of four different learning methods on the three datasets are shown in Table IV. From Table IV, we can see that the mean accuracies for all of concepts of our approach outperform other methods on the three datasets, and have been improved about 6%. In addition, the method "ID" has the poorest performance, and this is because this method ignores the relationship between training samples.

Fig. 7 also gives that certain results are worse for certain types of content, for example, Dancing on news video dataset. This is because the other three classes are very easy to be classified as this type of event, which leads to a high false alarm rate using "SVM". From the mean accuracies for all of the concepts, we can see that our FGSSMIL performs better than other methods. The reasons can be summarized as follows: 1) The entire video contains not only events of our interest, but also some clutter noises, which harms the classifier training and results in the bad performance. The increasing performance of FGSSMIL clearly illustrates the importance of temporal event localization in the training data. In addition, it is very suitable to formulate the event analysis as a multiple instance learning problem. 2) By combining large-scale unlabeled data, the FGSSMIL algorithm is effective to mine useful information. 3) Instance discrimination supplies some useful information for event modeling.

Compared with [10] as shown in Table II, the performance of our method FGSSMIL has some improvement. This is because the instance discrimination information is effective and we obtain a good solution by use of nonnegative multiplicative updating rule. Moreover, our method reduces the computational cost. On movie dataset, it costs about 4 h using the method in [10]. However, our method just needs about 23 min. On sports dataset, the computational cost is about 4.3 h in [10], which is more inefficient than our method with about 18 min. On news dataset, the computational time is about 3.7 h and 9 min, respectively. Therefore, the comparison demonstrates our proposed method is efficient and effective.

H. Video Annotation Evaluation

In this experiments, we apply the proposed FGSSMIL algorithm described above to temporally localize event boundaries on the three video datasets. To make a quantitative results, we adopt the localization accuracy as the evaluation criterion. The temporal localization accuracy is measured by the percentage of clips with relative temporal overlap to ground truth event segments greater than 0.3. This relatively loose threshold of 0.3 is used in order to compensate for the fact that temporal boundaries of events are somewhat ambiguous and not always accurately defined. Using this performance measure, we conduct the experiments for videos without text information alignment.

To obtain the ground truth event boundaries, we manually label 117, 114, and 105 video clips on movies, news, and sports video datasets, respectively. Then, we use these datasets to evaluate our proposed FGSSMIL algorithm for event localization. Our testing and training videos do not share the same scenes or actors. For the news video dataset, the results are shown in

TABLE V
COMPARED RESULTS OF THE PERFORMANCE OF LOCALIZATION ON NEWS DATASET

Event	Demonstration	Dancing	Eating	PlayingInstrument
# Positive Sample	31	28	23	32
ID	12.9%	10.7%	8.7%	12.5%
mi-SVM	51.6%	53.6%	60.8%	56.3%
GSSMIL	61.2%	60.7%	65.2%	59.3%
FGSSMIL	**67.74%**	**64.3%**	**65.2%**	**62.5%**

TABLE VI
COMPARED RESULTS ABOUT THE AVERAGE PRECISION OF LOCALIZATION ON THE THREE DATASETS

Dataset	Movie	News	Sports
# Positive Sample	117	114	105
ID	10.3%	11.2%	22.9%
mi-SVM	43.6%	55.6%	47.6%
GSSMIL	46.2%	61.6%	53.3%
FGSSMIL	**48.7%**	**62.3%**	**55.7%**

Table V. For video without text information, the average accuracy for event localization is only 11.2% with the method "ID". This is because the method "ID" just uses instance discrimination information, and ignores other useful information, such as, instance label constraints by bag information, similarities between instances. As a result, it is difficult to localize the event boundaries well. As for the FGSSMIL, it correctly localizes event boundaries, and has an average accuracy of 64.9% (labeled as "FGSSMIL"). However, if we use mi-SVM, the precision is only 55.6% (labeled as "mi-SVM"). The compared result shows that the localization performance is improved by using of unlabeled data. The average precision of event localization is shown in Table VI, which shows our algorithm can effectively localize event boundaries. Some automatically localized segments are shown in Fig. 6.

VI. CONCLUSIONS AND DISCUSSIONS

Video annotation is very important for content based video indexing, retrieval, and summarization. Therefore, we propose a generic framework to annotate videos from different video domains via semi-supervised learning to combine labeled data and unlabeled data. In this framework, the video annotation is formulate as an FGSSMIL algorithm, which has the following advantages. To alleviate human labeling effort, text information is mined and served as labels to assist model training. Besides the expert-level labels, the Internet is also able to provide a huge amount of event-relevant data (used as unlabeled data) to tackle the insufficient labeled data problem. Our FGSSMIL method can exploit both datasets together. To handle the ambiguity of event boundary in our labeled and unlabeled datasets, the FGSSMIL algorithm incorporates a multiple instance learning module. Moreover, the FGSSMIL algorithm makes use of the instance discrimination information and employs an effective method to describe the sample affinity, which is proved to boost the event recognition and localization performance significantly. Finally, to solve the optimum problem efficiently, CCCP and nonnegative multiplicative updating rule is adopted. In the future, we will extend the generic method to more broad categories and more video domains. In addition, we will research how to spatially localize event boundaries for video annotation.

Compared with the traditional graph-based methods for video annotation, our proposed graph-based semi-supervised multiple instance learning method is much more suitable for video annotation. This is because our proposed graph-based method makes use of the multiple instance learning property to localize the boundaries of the events. Moreover, our method learns optimal graph weights, and therefore it is capable of obtaining good performance. The drawback of our proposed model is that it only describes the temporal localization of an event, and does not take into account the spatial localization of an event. In the future, we will extend the model to localize the temporal and spatial boundaries of the events together.

REFERENCES

[1] Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/youtube.
[2] N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi, "Personalized abstraction of broadcasted American football video by highlight selection," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 575–586, 2004.
[3] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan, "Live sports event detection based on broadcast video and web-casting text,," in *Proc. ACM Multimedia*, 2006, pp. 221–230.
[4] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 68–75, 2002.
[5] J. Tang, X.-S. Hua, G.-J. Qi, Y. Song, and X. Wu, "Video annotation based on kernel linear neighborhood propagation," *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 620–628, 2008.
[6] X. Zhu, Semi-Supervised Learning Literature Survey University of Wisconsin-Madison, Madison, WI, Computer Sciences Tech. Rep. 1530, 2008.
[7] M. Fleischman and D. Roy, "Grounded language modeling for automatic speech recognition of sports video," in *Proc. ACL-08: HLT*, 2008.
[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. NIPS*, 2002.
[9] H. Cheng, Z. Liu, and Z. Liu, "Sparsity induced similarity measure for label propagation," in *Proc. ICCV*, 2009.
[10] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu, "A generic framework for event detection in various video domains," in *Proc. ACM Multimedia*, 2010.
[11] R. Yan and M. R. Naphade, "Semi-supervised cross feature learning for semantic concept detection in videos," in *Proc. CVPR*, 2005.
[12] J. Tang, X.-S. Hua, M. Wang, Z. Gu, G.-J. Qi, and X. Wu, "Correlative linear neighborhood propagation for video annotation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 409–416, 2009.
[13] T. Kato, H. Kashima, and M. Sugiyama, "Robust label propagation on multiple networks," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 35–44, 2009.
[14] A. Singh, R. D. Nowak, and X. Zhu, "Unlabeled data: Now it helps, now it doesn't," in *Proc. NIPS*, 2008.
[15] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *Proc. Int. Conf. Machine Learning*, 2009.
[16] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, "Image annotation by KNN-sparse graph-based label propagation over noisily tagged web images," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 2, no. 2, pp. 111–126, 2011.
[17] H. Buxton, "Learning and understanding dynamic scene activity: A review," in *Proc. Image and Vision Computing*, 2003.
[18] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern.*, 2004.
[19] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008.
[20] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar, "Movie/script: Alignment and parsing of video and text transcription," in *Proc. ECCV*, 2008.
[21] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *Proc. ICCV*, 2009.

[22] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in *Proc. ACM Multimedia*, Los Angeles, CA, 2000.
[23] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 796–807, Jul. 2003.
[24] D. Zhang and S. Chang, "Event detection in baseball video using superimposed caption recognition," in *Proc. ACM Int. Conf. Multimedia*, 2002.
[25] J. Wang, C. Xu, E. Chng, K. Wan, and Q. Tian, "Automatic generation of personalized music sports video," in *Proc. ACM Int. Conf. Multimedia*, 2005.
[26] J. G. Kim, H. S. Chang, K. Kang, M. Kim, J. Kim, and H. M. Kim, "Summarization of news video and its description for content-based access," *Int. J. Imag. Syst. Technol.*, vol. 13, no. 5, pp. 267–274, 2003.
[27] C. Huang, W. Hsu, and S. Chang, Automatic Closed Caption Alignment Based on Speech Recognition Transcripts, Columbia Univ., New York, Tech. Rep. 007, 2003.
[28] A. G. Hauptmann and M. J. Witbrock, "Story segmentation and detection of commercials in broadcast news video," Advances in Digital Libraries, 1998.
[29] T. Zhang, H. Lu, and S. Li, "Learning semantic scene models by object classification and trajectory clustering," in *Proc. CVPR*, 2009.
[30] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Proc. CVPR*, 2008.
[31] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. CVPR*, 2010.
[32] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *Proc. ICCV*, 2009.
[33] Y. Jia and C. Zhang, "Instance-level semisupervised multiple instance learning," in *AAAI'08: Proc. 23rd National Conf. Artificial Intelligence*, 2008.
[34] C. Wang, L. Zhang, and H.-J. Zhang, "Graph-based multiple-instance learning for object-based image retrieval," in *MIR'08: Proc. 1st ACM Int. Conf. Multimedia Information Retrieval*, 2008.
[35] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy c automatic naming of characters in tv video," in *Proc. BMVC*, 2006.
[36] M.-S. Dao and N. Babaguchi, "Sports event detection using temporal patterns mining and web-casting text," in *AREA'08: Proc. 1st ACM Workshop Analysis and Retrieval of Events/Actions and Workflows in Video Streams*, 2008.
[37] [Online]. Available: http://www.dtSearch.com.
[38] T. Dietterich, R. Lathrop, and T. Lozano-Perez, Solving the Multiple-Instance Problem With Axis Parallel Rectangles, 1997, Artificial Intelligence.
[39] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. NIPS*, 2002.
[40] A. Smola, S. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *Proc. Int. Workshop Artificial Intelligence and Statistics*, 2005.
[41] A. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Computat.*, vol. 15, no. 4, pp. 915–936, 2003.
[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[43] [Online]. Available: http://www.mosek.com/.
[44] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. PETS*, 2005.
[45] M. Müller, *Information Retrieval for Music and Motion*. New York: Springer, 2007, p. 65.
[46] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, 1980.
[47] [Online]. Available: http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations.

Tianzhu Zhang (M'11) received the Bachelor's degree in communications and information technology from Beijing Institute of Technology, Beijing, China, in 2006 and the Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2011.

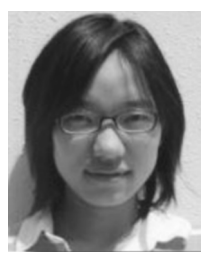
He is a postdoctoral fellow with the Advanced Digital Sciences Center (ADSC), Singapore. He does extensive research on computer vision and multimedia, such as action recognition, video surveillance, and object tracking.

Changsheng Xu (M'97–SM'99) is a Professor in the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and Executive Director of China-Singapore Institute of Digital Media. His research interests include multimedia content analysis/indexing/retrieval, pattern recognition, and computer vision. He has held 30 granted/pending patents and published over 200 refereed research papers in these areas.

Dr. Xu is an Associate Editor of *ACM Transactions on Multimedia Computing, Communications and Applications* and *ACM/Springer Multimedia Systems Journal*. He served as Program Chair of ACM Multimedia 2009. He has served as associate editor, guest editor, general chair, program chair, area/track chair, special session organizer, session chair, and TPC member for over 20 IEEE and ACM prestigious multimedia journals, conferences, and workshops.

Guangyu Zhu received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include image/video processing, multimedia content analysis, computer vision and pattern recognition, and machine learning.



Si Liu is pursing the Ph.D. degree in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

She is currently a Research Assistant at the Learning and Vision Group of the National University of Singapore. Her research interests include computer vision and multimedia.

Hanqing Lu (SM'06) received the Ph.D. degree from Huazhong University of Sciences and Technology, Wuhan, China, in 1992.

Currently, he is Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include image similarity measure, video analysis, object recognition, and tracking. He published more than 200 papers in those areas.