On the Annotation of Web Videos by Efficient Near-duplicate Search

Wan-Lei Zhao, Xiao Wu, *Member, IEEE*, and Chong-Wah Ngo, *Member, IEEE*

***Abstract*—With the proliferation of Web 2.0 applications, user-supplied social tags are commonly available in social media as a means to bridge the semantic gap. On the other hand, the explosive expansion of social web makes overwhelming number of web videos available, among which there exists a large number of near-duplicate videos. In this paper, we investigate techniques which allow effective annotation of web videos from a data-driven perspective. A novel classifier-free video annotation framework is proposed by first retrieving visual duplicates and then suggesting representative tags. The significance of this paper lies in the addressing of two timely issues for annotating query videos. First, we provide a novel solution for fast near-duplicate video retrieval. Second, based on the outcome of near-duplicate search, we explore the potential that the data-driven annotation could be successful when huge volume of tagged web videos is freely accessible online. Experiments on cross sources (annotating Google videos and Yahoo! videos using YouTube videos) and cross time periods (annotating YouTube videos using historical data) show the effectiveness and efficiency of the proposed classifier-free approach for web video tag annotation.**

I. Introduction

Despite the advance in content analysis of videos, overcoming the semantic gap between human perception and low-level visual features remains a challenging problem. Existing approaches in video annotation depend heavily on the machine learning techniques (e.g., Support Vector Machines) to map the low-level features to high-level semantic concepts. These so-called model-based approaches normally involve the learning of a large set of concept classifiers for labeling the incoming data. In general, these approaches are not competent for managing the ever-increasing number of web videos due to the following two main reasons:

- A large amount of balanced labeled samples is often required for effective classifier learning. Nevertheless, the scarcity of training examples commonly exists in many applications. Collecting a large set of noise-free training examples with sufficient positive samples for learning is always not easy. Manual annotation of training examples can be laborious, and most labeling efforts are indeed spent in annotating negative examples. Considering these difficulties, building a large set of classifiers scalable for annotating most of the concepts in web videos is beyond the current state-of-the-art technologies.
- The size of vocabulary is huge and the meaning of concepts may change dynamically. In social media, for example, a word may evolve over time and can change according to context. Novel words or phrases may emerge when new topics are being discussed. Learning classifiers in such scenario is difficult to cope with and completely model the evolving nature of web environment.

The emergence of Web 2.0 technology makes video a popular social media shared among web users. Some of these videos come alongside with tags which provide semantics and context about the video content. An intuitive idea for annotation is by utilizing existing tags to label new videos. Basically, given an un-tagged video, similar videos are first retrieved from database. The associated tags of similar videos are examined and then appropriate tags are picked for annotating the new video. From the data perspective point of view, such data-driven approaches are possible when there are enough videos and tags available to characterize any new incoming data. This paves a new way of annotation through a model-free data-driven methodology. Such techniques have recently been evident in [15], [19], [24], [26], which are also referred to as “annotation by search”. In [24], a large dataset of 80 million tiny images is collected for object and scene recognition by nearest neighbor search. In [15], [26], content-based retrieval techniques based on global features are exploited for image annotation. In [19], tag propagation technique is developed by crawling tags of similar videos for annotation by using text and global visual features. Similar in spirit, this paper also explores search-based annotation by the nearest neighbor search of examples in large visual corpus. Different from previous works, our work is based on effective and efficient near-duplicate video retrieval by using local keypoints, targeting for annotating web videos. Compared to similarity-based labeling on images and videos as in [15], [19], [24], [26], near-duplicates searched by local keypoints provide more reliable and accurate information for video annotation. Keypoint based video search, different from global features, requires the considerations of point matching, geometric checking and segment localization, which have not

The work described in this paper was fully supported by a grant from City University of Hong Kong (Project No. 7002438).

Wan-Lei Zhao is with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong (e-mail: wzhao2@cs.cityu.edu.hk).

Xiao Wu is with the Department of Computer Science and Engineering, Southwest Jiaotong University, No. 111, North Section 1, 2nd Ring Road, Chengdu, China, and the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong (e-mail: wuxiaohk@home.swjtu.edu.cn).

Chong-Wah Ngo is corresponding author and is with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong (e-mail: cwngo@cs.cityu.edu.hk Tel: 852-2784-4390 Fax: 852-2788-8614).

yet been explored by other search-based annotation techniques.

In this paper, we investigate two main issues for data-driven video annotation: (1) how to efficiently search for similar videos by local features, and (2) how effectively the tags from similar videos can be recycled for tagging. We consider the similar videos as the set of duplicate or partial duplicate videos which commonly exist in most social media websites such as YouTube. For the first issue, we propose an efficient near-duplicate video retrieval framework by considering three aspects: indexing of local visual features, fast pruning of false matches at frame-level, and localization of near-duplicate segments at video-level. For the second issue, a weighted majority voting scheme is adopted for tag recommendation by studying the tagging behaviors in the pool of retrieved near-duplicates. Intuitively, if different users label visually similar videos with the same tags, these tags are likely to reflect an objective view of the video content. The premise of the proposed methodology is that there exists at least one near-duplicate video in the corpus. Our approach cannot provide tag if no near-duplicate videos can be found in the reference set.

We conduct experiments on a large-scale video dataset and consider two tasks: *cross-source tagging* which uses existing tags in one search engine to tag videos available in other search engines, and *cross-time tagging* which uses existing tags collected over years to tag recently uploaded videos. Both types of tagging have their own applications. We regard cross-source tagging as an economic way to propagate the metadata among search engines, so as to improve the retrieval performance by enriching or exchanging the tags among different engines. Cross-time tagging, by suggesting tags to newly uploaded videos, can enhance the retrieval of similar videos manipulated by various parties which may carry new or additional message over time along the manipulation history.

The rest of this paper is organized as follows. Section II gives a brief overview of related work. Section III introduces the proposed framework for model-free web video annotation. Section IV outlines the efficient algorithm for near-duplicate search, while section V presents data-driven tag annotation. Section VI describes our experimental setup for cross-source and cross-time tagging, and section VII further details our empirical findings. Finally, section VIII concludes this paper.

II. Related Work

A. Tag Annotation

The performance of semantic-based image/video search depends largely on the quality of the keywords or annotation. To bridge the semantic gap between low-level visual features and semantic concepts, image auto-annotation and object recognition have attracted the interest of researchers in recent years. Many learning models (e.g., [20]) have been proposed to automatically assign keywords onto images or image regions. For automatic image annotation, the works can be categorized into two directions: to learn the conditional probability, or to learn the joint probabilities between images and words. Unfortunately, the performances of these statistical models are still far from being acceptable for practical applications. Instead, social tagging is widely adopted in various social media websites such as del.icio.us, Flickr, and YouTube. The descriptive metadata generated by grass-root users are often exploited for effective organization of web resources.

Existing works on tagging services cover a wide range of research topics, including resolving tag ambiguity [27], analyzing usage patterns of tagging systems [5], mining social interests through tags [14], automating tag assignment [2], [16], [21], [23], [28], and so on. Among them, there have been numerous efforts on automatic tag suggestion or recommendation [2], [15], [21], [23], [26]. A common strategy is to suggest tags based on personal history, geographic location and time [1]. Co-occurrence of tag is a vivid clue that has been explored by [21], [27]. In [21], tag aggregation algorithm is proposed to rank recommended tags according to tag co-occurrence, frequency, and long-tail distribution effect. In [27], a measure is proposed to determine the ambiguity of a tag set, and new tags that can disambiguate the original tags are suggested. In addition to tag recommendation, tag refinement is also explored to prune or re-rank tags. Ontology such as WordNet is employed for pruning semantically irrelevant tags [31]. Tags are refined by re-ranking candidate tags using random walk [25]. A graph with tags as vertices and tag co-occurrences as edges is constructed to rank tags according to their popularity. Recently, a neighbor voting algorithm is proposed for image retrieval [16], which predicts the relevance of user-contributed tags. By taking into account the tag and visual correlation, the recent work in [28] formulates tag recommendation as a learning problem.

Search-based annotation has also captured numerous research attention recently. In [15], [26], a corpus of 2.4 million images are crawled from web for image annotation. A high-dimension indexing scheme (namely Multi-Index) and a search result clustering technique (namely SRC) are proposed in [15] for annotation through large-scale search. A more sophisticated divide-and-conquer framework which considers text and visual search is later developed in [26] for data-driven tagging. In [19], variants of graph reinforcement algorithm are proposed for propagating tags from similar documents to query videos. These works [15], [19], [26] consider only global visual features for search. In [19], [26], initial textual keywords or labels are further assumed to be available to guarantee efficient search and effective propagation. While similar in spirit, our works in this paper are different from [15], [26] in several aspects. First, we do not assume the availability of textual keywords to initiate the search for annotation. Instead, the search is purely based on visual information. To ensure the robustness of visual search, we consider local features which are very different from global features such as color moment and edge histogram used in [15], [26]. Second, we consider scalable search of partial near-duplicate videos, where spatial geometric and temporal consistency information are taken into account. The works in [15], [26] which perform image search by global visual features thus are not directly extensible to our work. Similarly, the work in [19] aims for effective propagation of tags from similar videos but scalability is not considered. Thus, extending the propagation algorithms to thousands of web videos remains a challenging issue in

general.

B. Near-Duplicate Video Retrieval

Near-duplicate videos are identical or approximately identical videos with similar appearance, but varying in terms of formatting, encoding parameters, editing, photometric variation, viewpoints, and change in camera parameter or setting [29], [33]. Existing works on near-duplicate retrieval can be broadly grouped into two categories. One category demands speedy response while the other emphasizes more on detection effectiveness. The first category aims for rapid retrieval (e.g., [6], [32]) and thus global features derived from color and ordinal signature are popularly employed. These approaches are highly suitable for identifying near identical videos. For videos with partial duplicate, either spatially or temporally, global features are known to be less reliable.

The second category addresses the robustness issue by mainly employing local point features [17]. Local points (keypoints) are salient local patches detected over different scales. Its effectiveness has been demonstrated by various works (e.g., [9], [12], [13], [33], [34]), where near-duplicates with considerable changes in background, color and lighting can be successfully identified. While these approaches are robust in general, the robustness comes with the expense of computational cost. To tackle this problem, different approaches have been proposed. Locality sensitive hashing (LSH) is adopted in [12], while a distortion-based probability similar search algorithm based on LSH is proposed for fast duplicate search in [10]. Another popularly adopted technique is retrieval by visual keywords (VK) [22], also known as bag-of-words. Under this technique, keypoints are quantized into groups (dictionary) and each group (an entry of dictionary) is viewed as a word. As a result, instead of representing video content with hundreds to thousands of keypoint features in high dimension (e.g., 128 dimensions for SIFT descriptor [17]), VK characterizes video content as a histogram of words which facilitates fast matching.

VK histogram is often a sparse feature vector in high dimensional space (e.g., 10,000 words). Thus, inverted file index is employed for fast matching of visual words [22]. A major weakness of VK is visual ambiguity caused by keypoint quantization. Specifically, large (small) dictionary leads to miss (false) matching. Several approaches have been proposed to address this problem, for instance, Hamming embedding (HE) [7], soft-weighting (SW) [8] and weak geometric consistency (WGC) [7]. HE keeps a bit-pattern signature for each visual word for pruning false positives due to histogram matching, while SW assigns multiple words to keypoints to resolve visual ambiguity. WGC is a weighting scheme which re-ranks the quality of histogram matches by checking their geometry consistency. VK normally operates on keyframe-level. Specifically one keyframe is represented by a histogram. To match two videos, Hough Transform (HT) [4] is another technique often employed to measure the degree of match between videos. HT considers time lags between the matched keyframes from two videos. The time lags are utilized as the cue to measure video similarity.

Annotating web videos expects timely response. On the other hand, quality of web search will impact the result of annotation. Choosing appropriate technique is thus a trade-off between retrieval speed and accuracy. In this paper, we adopt VK+HE together with WGC and HT for scalable retrieval. Both WGC and HT are revised for enhancing search accuracy, leading to a much better performance compared to their original versions.

III. Annotation by Search

A. Data-driven Annotation

The problem of annotating web videos can be formulated as finding a group of tags which maximize the conditional distribution $p(t|V_i)$:

$$t^* = \underset{t}{\operatorname{argmax}}\, p(t|V_i), \tag{1}$$

where V_i is a web video to be annotated and t^* is a pool of candidate tags. According to Bayesian rule, Eqn. 1 can be expanded to:

$$t^* = \underset{t}{\operatorname{argmax}} \sum_k p(t|V_k)p(V_k|V_i). \tag{2}$$

Intuitively, for a web video to be annotated, t^* appears more frequently in the contexts of similar videos than dissimilar ones. Hence, we can approximate Eqn. 2 by generating t^* from similar videos instead of the whole video corpus. Denote Θ_i as the set of similar videos. The problem of annotation becomes equivalent to searching Θ_i and collecting most probable tags from Θ_i. Eqn. 2 can then be reformulated as:

$$t^* = \underset{t}{\operatorname{argmax}}\, p(t|\Theta_i)p(\Theta_i|V_i), \tag{3}$$

where $p(\Theta_i|V_i)$ acts as the search process to identify similar videos, and $p(t|\Theta_i)$ represents the tag generation process. Based on the equation, a two-step solution for data-driven web video annotation is proposed:

- **Scalable search**: retrieving a collection of near-duplicate videos Θ_i, and
- **Tag annotation**: mining annotations t^* from the tags associated with videos Θ_i.

B. Framework

Figure 1 illustrates the proposed framework for both cross-source and cross-time tagging. Two major components in this framework are: efficient near-duplicate search and tag annotation. For offline indexing, videos are first crawled from web to form a large corpus. These videos are pre-processed by performing shot boundary detection and then keyframe selection. Local keypoints are extracted from the keyframes and clustering is carried out to quantize the keypoints into a visual dictionary. Each keypoint in the keyframes is then encoded with a visual word in the dictionary, and this forms a bag of words for each keyframe. Inverted file indexing plus Hamming embedding [7] is employed to support scalable keyframe retrieval with fast similarity evaluation. Similar procedure is applied to the given queries.

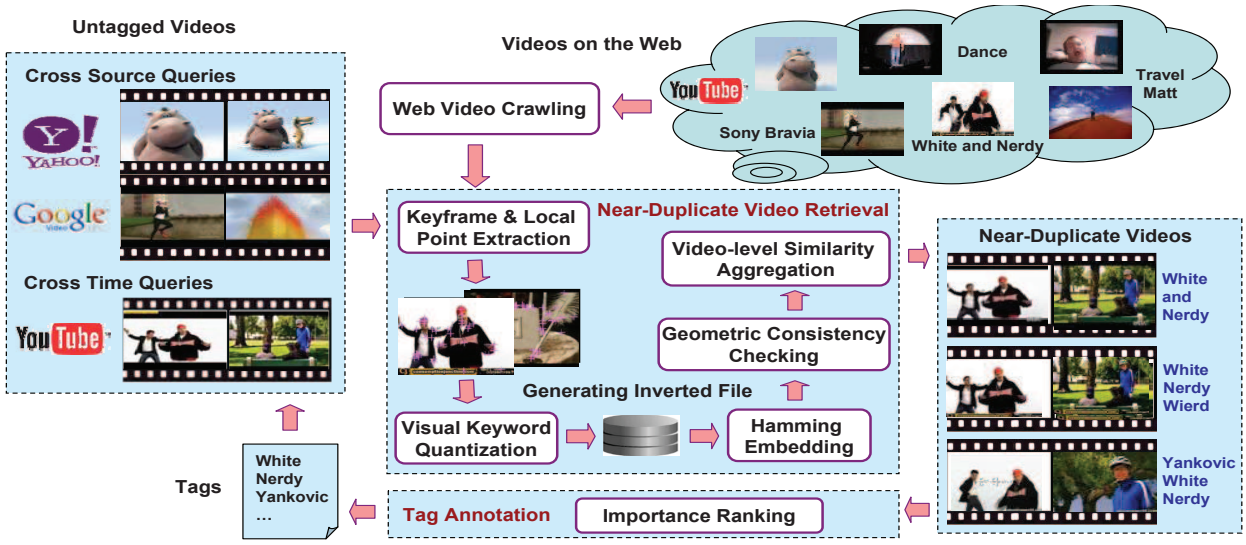


Fig. 1. Framework for data-driven web video annotation.

In near-duplicate search, the keyframes which are similar to the query keyframes are retrieved from the video corpus via visual keywords and inverted file (Section IV-A). The retrieved keyframes are further re-ranked according to their geometric consistency with the query. An efficient evaluation based on weak geometric consistency checking is proposed (Section IV-B). Finally, the similarity of a video is determined by aggregating the scores of keyframes in the video and weakly considering their temporal consistency with the query. The video-level similarity aggregation is based on 2D Hough Transform together with a proposed reverse-entropy measure (Section IV-C).

In tagging, the collection of candidate tags is pooled from the set of retrieved videos. An effective measure which considers tag frequency, the number of tags, and the similarity weight of videos is proposed to rank the tags according to their relevance (Section V). Finally, the first few tags with higher rank are recommended for annotating the query videos.

IV. Scalable Near-duplicate Video Retrieval

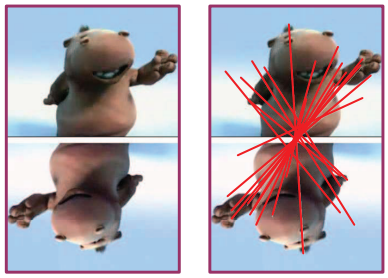
A. Visual Keywords (VK) and Inverted File Indexing

To ensure a reliable retrieval, we adopt local keypoint descriptors as the features for near-duplicate retrieval. Nonetheless, the number of keypoints in a keyframe can range from hundreds to thousands, while the dimension of descriptor is typically high. Matching keypoints between two keyframes becomes extremely slow. Thus, we employ clustering approach by first quantizing keypoints into a visual dictionary (codebook). Each entry in the dictionary (or centroid of a cluster) corresponds to a word. By mapping each keypoint in a frame to the nearest word, this forms a bag-of-words, which is represented in the form of histogram, describing the visual content of the keyframe. Each bin in the histogram accumulates the number of words found in the keyframe. Measuring the similarity between two keyframes is then performed by bin-to-bin matching of their histograms. Denote m as the vocabulary size of words, and $f_k(I_i)$ as the weight of *kth* bin in keyframe I_i, we use *cosine similarity* to measure the closeness between keyframes I_i and I_j:

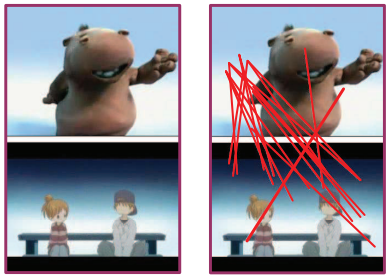
$$sim_{ij} = \frac{\sum_{k=1}^{m} f_k(I_i) \times f_k(I_j)}{\sqrt{\sum_{k=1}^{m} f_k(I_i)^2 \sum_{k=1}^{m} f_k(I_j)^2}}. \quad (4)$$

To ensure the coverage of dictionary, the number of words is usually large (e.g., $\geq$ 10,000). Directly matching two histograms using Eqn. 4 will not be extremely fast in this case. Nevertheless, since the histogram is normally very sparse, the matching can be efficiently conducted by exploiting structure such as inverted file index [22] which is popularly used in text information retrieval. The index stores the keyword-image relationship, in which each entry (or row) corresponds to a keyword and links to the list of keyframes which contain the word. As a consequence, given a keyframe, the words are hashed into the index and the matched keyframes are retrieved. Cosine similarity is thus only evaluated for a subset of keyframes in the dataset and for those non-zero entries in the histograms.

We adopt two techniques: 2-level vector quantization (VQ) and Hamming embedding (HE) [7] to further speed up the online retrieval time. Multiple-level VQ allows efficient encoding of keypoints to keywords without exhaustive search of the nearest words. To reduce the information loss caused by VQ, HE maintains a binary signature for each keypoint. The signature is indexed in the inverted file to facilitate the measurement of keypoint distances for keypoints falling into the same visual word. During retrieval, any two matched visual words can be pruned if the Hamming distance between their signatures is large. This results in less words being involved in similarity measuring and also the subsequent steps of geometric checking. In our implementation, we choose 32-bit binary signature. The threshold for Hamming distance is set at 15 such that any matched visual words whose distance exceeds this value will be pruned. In our experiment, querying a keyframe against a dataset of 632,498 keyframes only requires 0.08 seconds to complete.



(a) Near-duplicate keyframes.



(b) Dissimilar keyframes.

Fig. 2. Two pairs of keyframes (left) with similar matching scores based on Eqn. 4. Portion of the matched visual words are highlighted (right).

B. Keyframe-level Geometric Consistency Checking

The keypoint quantization introduces ambiguity in visual matching. For example, words from the same bin are always matched regardless of their actual distance measured by keypoint descriptors. For words from large clusters, this could cause excessive number of false matches. Thus geometric consistency checking is a post-processing step aiming to examine the coherency of matches between two sets of visual words. Figure 2 shows an example that two pairs of keyframes have similar matching scores as computed by Eqn. 4. However, visually the keyframes in Figure 2(b) are very different. Ideally, by recovering their underlying geometric transformation from the word matches as shown in Figure 2, the dissimilar keyframes can be pruned.

*1) **Weak Geometry Consistency (WGC)**:* Recovery of transformation is often done by RANSAC [18]. However, such estimation is always costly and not appropriate when large number of keyframes are required to be investigated. WGC [7] is a recently proposed technique which exploits the weak or partial geometric consistency without explicitly estimating the transformation by checking the matches from one keyframe to another. Given two matched visual words $p(x_p, y_p)$ and $q(x_q, y_q)$ from two keyframes respectively, WGC estimates the transformation from p to q as:

$$\begin{bmatrix} x_q \\ y_q \end{bmatrix} = s \times \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \quad (5)$$

where (x_p, y_p) and (x_q, y_q) are the 2D spatial positions of p and q in x-y coordinate. In Eqn. 5, there are three parameters to be estimated: the scaling factor s, the rotation parameter θ and the translation $[T_x, T_y]^t$. In WGC, only the parameters s and θ are estimated. For efficiency, the scale and rotation can be derived directly from the local patches of p and q without the explicit estimation of Eqn. 5. The scale s is approximated as:

$$\widetilde{s} = 2^{(s_q - s_p)}, \quad (6)$$

where s_p, s_q are the characteristic scales of words p and q respectively. The scale values of words are known by the time when their corresponding keypoints (or local patches) are detected. For instance, the value s_p indicates the scale level which p resides in the Laplacian of Gaussian (or Difference of Gaussian) pyramid [17], [18]. Similarly, the orientation θ is approximated as:

$$\widetilde{\theta} = \theta_q - \theta_p, \quad (7)$$

where θ_p and θ_q are the dominant orientations of visual words p and q estimated during keypoint detection [17].

WGC computes $log(\widetilde{s})$ and $\widetilde{\theta}$ for each matched visual word between two keyframes. By treating scale and rotation parameters independently, two histograms h^s and h^θ, referring to the scale and orientation consistency respectively, are produced. Each peak in a histogram means one kind of transformations being performed by a group of words. Ideally, a histogram with one or few peaks hints the consistency of geometry transformation for most visual words in the keyframes. WGC utilizes the consistency clue to adjust the similarity of keyframes computed in Eqn. 4 by:

$$sim_{wgc}(i,j) = min(max(h^s), max(h^\theta)) \times sim_{ij}. \quad (8)$$

The similarity is boosted, by a factor corresponding to the peak value in scale or orientation histogram, for keyframe pairs which show consistency in geometry transformation.

*2) **Enhanced Weak Geometry Consistency (E-WGC)**:* The merit of WGC lies in its simplicity and thus efficiency in transformation estimation. Nevertheless, such estimation is not always reliable. The main reason for unreliable estimation is due to the fact that the characteristic scale and dominant orientation estimated from keypoint detection are not always discriminative enough. For example, although DoG (Difference of Gaussian) detector adopts 5 levels of Gaussian pyramid for keypoint localization, most points are detected at level 1. As a consequence, the scale histogram always has a peak corresponding to level 1.

We propose the enhancement of WGC by also including translation information. Combining equations 5, 6 and 7, we have the WGC estimation as:

$$\begin{bmatrix} \widetilde{x_q} \\ \widetilde{y_q} \end{bmatrix} = \widetilde{s} \times \begin{bmatrix} \cos\widetilde{\theta} & -\sin\widetilde{\theta} \\ \sin\widetilde{\theta} & \cos\widetilde{\theta} \end{bmatrix} \times \begin{bmatrix} x_p \\ y_p \end{bmatrix}. \quad (9)$$

Deriving from equations 5 and 9, the translation τ of the visual word q can be efficiently estimated by

$$\tau = \sqrt{(\widetilde{x_q} - x_q)^2 + (\widetilde{y_q} - y_q)^2}. \quad (10)$$

Ideally, the matched visual words which follow consistent transformation should have similar values of τ, and thus τ can be used to directly adjust the keyframe similarity as in Eqn. 8. There are two advantages with this simple scheme. First, the inclusion of translation information provides another geometric clue in addition to scale and rotation. Second, Eqn. 10 has jointly integrated the clues from scale, rotation and translation, thus generating one histogram of τ is enough

for similarity re-ranking. In addition, Eqn. 10 can be trivially computed without incurring additional computational cost. Since Eqn. 10 is an enhanced version of equations 6 and 7, we name our approach *enhanced* WGC, or E-WGC in short.

In E-WGC, only one histogram is generated based on the values of translation τ. Similar to WGC, histogram peak is located to re-rank the keyframe similarity. For robustness, the peak is smoothed by considering the moving average of two neighboring bins. As a result, the value of peak is computed as:

$$\tau_{peak} = |h_i| + |h_{i-1}| + |h_{i+1}| - 2 \times \frac{\sum_{j=1}^{m} |h_j|}{m}, \qquad (11)$$

where h_i is the bin with peak value, and *m* is the number of histogram bins. Ultimately, the keyframe is re-ranked as:

$$sim_{ewgc}(i,j) = (\frac{\tau_{peak}}{M_{vk}})^{\gamma} \times sim_{ij}, \qquad (12)$$

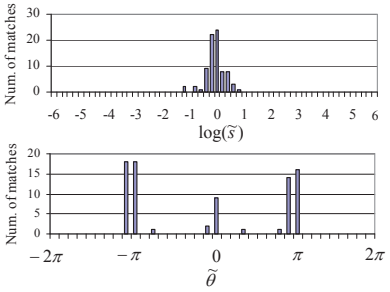
where τ_{peak} is normalized by M_{vk} which denotes the number of matched visual words in two keyframes. We amplify the ratio of τ_{peak} to M_{vk} by a factor of γ so as to increase the gap between similar and dissimilar keyframes. The factor γ is a parameter empirically set equal to 3, which will not affect the re-ranking result. We include this factor for the purpose of selecting only few most similar videos for tagging, which will be further elaborated in Section V.

Figure 3 compares WGC and E-WGC. Figures 3(a)-(b) show the scale and rotation histograms of WGC from the matches of visual words in Figures 2(a) and 2(b) respectively. Figures 3(c)-(d) show the translation histograms[1] of E-WGC in Figure 2. For WGC, despite that the keyframes in Figure 2(a) are the rotated version of one another, there are two peaks found in the difference of $\widetilde{\theta}$ histogram of 3(a). Similarly, by observing the histograms in Figure 3(b) computed from the keyframes in Figure 2(b), there are apparent peaks in both histograms, though the keyframes in Figure 2(b) are dissimilar. As a result, the re-ranking scheme in Eqn. 8 incorrectly boosts their similarity in this case. In contrast, as seen in Figures 3(c)-(d), the translation histogram of E-WGC shows an apparent peak for the near-duplicate pair in Figure 2(a), while there is no peak with high score for the dissimilar keyframes in Figure 2(b). The re-ranking formula in Eqn. 12 therefore boosts the similarity of Figure 2(a) but not Figure 2(b). Compared to WGC, E-WGC can more effectively distinguish near-duplicates from dissimilar keyframes.

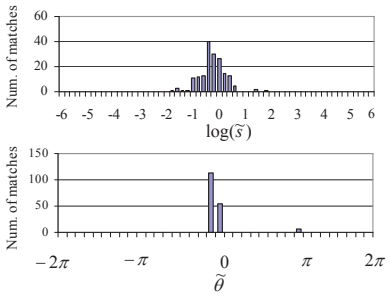
C. Video-Level Similarity Aggregation

Similarity aggregation involves measuring the sequence similarity for videos where their keyframes are fully or partially matched to the keyframes of a query video. Given a set of keyframe pairs from a video V_k and a query *Q*, the similarity between V_k and *Q* can be counted by aggregating the number of keyframe matches. Such measure, nevertheless, does not consider temporal consistency and the noisy matches can be easily included in similarity counting. Hough Transform (HT) is a technique aiming to aggregate the keyframe matches

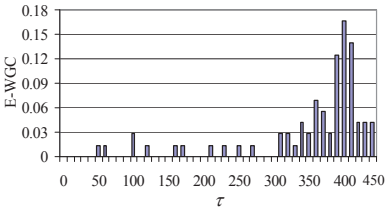
[1]Note that the histograms are normalized by M_{vk}.



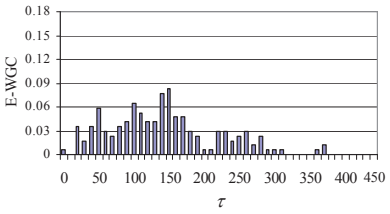
(a) WGC histograms for Fig.2(a).



(b) WGC histograms for Fig.2(b).



(c) E-WGC histogram for Fig.2(a).



(d) E-WGC histogram for Fig.2(b).

Fig. 3. Comparison between WGC and E-WGC: (a) and (b) are the scale and rotation histograms of WGC; (c) and (d) are the translation histograms for E-WGC. For WGC, there are obvious peaks for both near-duplicate and dissimilar keyframes. In E-WGC, an apparent peak is found only in the histogram of near-duplicate keyframes. Therefore, E-WGC is capable of distinguishing near-duplicate from dissimilar keyframes.

by weakly considering their temporal consistency [4]. HT is basically a voting scheme which accumulates scores from matches with similar time lags. Given a keyframe pair I_i and I_j with similarity score $sim_{ewgc}(i,j)$ as computed in Eqn. 12 and temporally located at time t_1 and t_2 of videos V_k and *Q* respectively, the time lag is computed as:

$$\delta_{i,j} = t_1 - t_2. \qquad (13)$$

HT aggregates the similarity score as a result of one keyframe match into a 2-dimensional histogram, with one dimension as the video ID and the other dimension as the time lag. In this histogram, video ID is a unique integer assigned to a video, and the range of time lag is quantized into bins by a bandwidth of δ_0. Each keyframe matching pair I_i and I_j contributes a score of $sim_{ewgc}(i,j)$ to the bin $[k,b]$, where *k* is

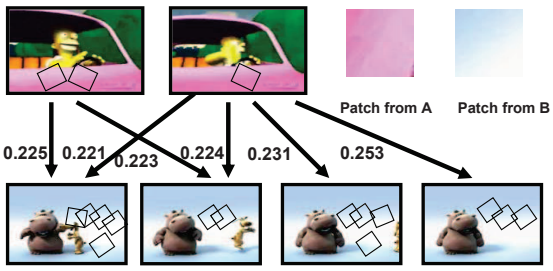


Fig. 4. Due to quantification error of visual words, a keyframe may be matched to multiple keyframes in another video. The numbers associated with edges are the similarity scores computed with Eqn. 4, and the patches denote the matched words

the video ID of V_k and b=$\lfloor \frac{\delta_{i,j}}{\delta_0} \rfloor$. Consequently, a peak in the histogram, in the form of a triple $[k, b, score_{kb}]$, corresponds to an accumulated score $score_{kb} = \sum_{i,j} sim_{ewgc}(i,j)$ of keyframes in V_k which are temporally aligned with the query video Q (i.e., b=$\lfloor \frac{\delta_{i,j}}{\delta_0} \rfloor$). In other words, peaks in the histogram hint the video segments which are similar to Q. Detecting the peaks is basically equivalent to finding the partial near-duplicates of Q.

Let $\mathcal{H}[k,b]$ be the 2D Hough histogram, where $1 \leq k \leq n$, and n is the number of videos having the matched keyframes with query Q. HT measures the similarity of video V_k to Q by

$$\mathcal{S}im_{ht}(V_k, Q) = max_b(\mathcal{H}[k,b]), \tag{14}$$

In other words, the similarity of two videos is determined by the maximum aggregated similarity score of keyframes from both videos which are consistently matched along the temporal dimension.

1) ***Reverse-Entropy Ranking****:* While Hough Transform is efficient to implement, it has the deficiency that the influence of noisy matches is not carefully tackled. As indicated in [4], the similarity aggregation is often mixed with considerable portion of false positive matches. The reasons are mainly due to two practical concerns. First, shot boundary detection is not always perfect. False detection can cause excessive number of shots (and keyframes) which are similar to each other within a video. As a consequence, this often results in one keyframe from a video being matched to several keyframes in another video or vice versa. Second, the imprecise matching of visual words due to quantization error, as well as E-WGC which only weakly considers the geometric transformation, also introduces random false matches. These practical concerns jointly make the similarity aggregation in HT lack of robustness. Figure 4 illustrates an example where the second keyframe from a video is matched to almost all the keyframes of another video. It causes the video pair to have high aggregated score according to Eqn. 14.

To solve this problem, we revise Eqn. 14 by taking into account the granularity of matching. The intuitive idea is that a keyframe which matches to multiple keyframes in another video is given less priority when determining video similarity. Thus, the aim is to lower the scores of video segments which include excessive matching. Let the 2D Hough bin $[k,b]$ as the peak which gives rise to the similarity between videos V_k and Q as in Eqn. 14, and assume that the bin corresponds to a segment S_v from V_k and another segment S_q from Q. Let N_q be the number of keyframes in S_q, and η_l be the number of keyframes from S_v with matches to the lth keyframe of S_q. We employ entropy to measure the associative mapping from S_q to S_v as

$$RE(S_q \rightarrow S_v) = \frac{-1}{\log Z}(\sum_{l=1}^{N_q} \frac{\eta_l}{Z} \times log\frac{\eta_l}{Z}), \tag{15}$$

where $Z = \sum_{l=1}^{N_q} \eta_l$ is the total number of keyframe matches between videos V_k and Q. The value of entropy ranges within $[0,1]$. The measure of entropy depends upon the granularity of matches from one video to another. Matching which exhibits one-to-one correspondence will receive the highest entropy value of 1. In contrast, for the cases of one-to-many or many-to-one matching, the entropy value will be low. A special case happens when only one keyframe in S_q has matches, and the keyframe matches to all the keyframes in S_v. In this case, the value of entropy will be 0. Since the definition of entropy here is different from the conventional definition where a value of 1 indicates uncertainty while a value of 0 indicates confident match, we name the entropy measure in Eqn. 15 as *Reverse-Entropy (RE)* measure.

The measure of RE is not symmetric, meaning that the matches for $S_q \rightarrow S_v$ will have a different RE value from that of $S_v \rightarrow S_q$. Thus the final value of RE is defined as:

$$RE(V_k, Q) = min(RE(S_q \rightarrow S_v), RE(S_v \rightarrow S_q)). \tag{16}$$

We use the RE measure in Eqn. 16 to estimate the similarity between videos V_k and Q as following:

$$\mathcal{S}im_{re}(V_k, Q) = \begin{cases} Sim_{ht}(V_k, Q) \times RE(V_k, Q)^2 & \text{if } RE(V_k, Q) \neq 0 \\ Sim_{ht}(V_k, Q) \times \frac{1}{\sqrt{Z}} & \text{Otherwise} \end{cases} \tag{17}$$

The original similarity is devalued by the square of RE to impose a heavier penalize on videos having noisy matching segments. Notice that when RE equals to 0, it indicates that only one keyframe of a video (either Q or V_k) is matched to keyframes of another video. In this case, the similarity is weighted by $\frac{1}{\sqrt{Z}}$ so as to avoid the similarity score from dropping abruptly v to zero.

D. Time and Space Complexity

While the framework involves a variety of components, the retrieval speed is highly efficient mainly due to the use of inverted file which is essentially a hashing technique. There are several factors which govern the time complexity of retrieval. These include the size of visual words (w), number of keyframes per query (m), and number of keypoints per keyframe (p).

Mapping a keypoint to a word and eventually retrieving the list of keyframes containing the word takes $O(w)$ time. The speed can be improved to $O(log(w))$ with the use of multi-layer vector quantization. Thus, mapping the keypoints from an entire query video to visual words for retrieval costs $O(mplog(w))$. The size of candidate videos retained for E-WGC and HT is query dependent. Assuming that there are

n keyframes being retrieved and each keyframe contains q keypoints, in the worst case, E-WGC will take $O(mnpq)$ to verify every matching point, and HT will take $O(mn)$ to check temporal alignment. Summing up all the components, the complexity is $O(mplog(w)) + O(mnpq) + O(mn)$. Let K be the total number of keyframes indexed in dataset. In practice, $m, n << K$, and the average number of keypoints (p and q) is typically about 260. Due to the use of HE, the number of keypoints to be verified by E-WGC can be reduced by another 70-85%. In brief, the time complexity is sub-linear to the number of keyframes and keypoints in a dataset, and in practice, the retrieval speed can be very efficient.

In terms of space complexity, each visual word in the inverted file stores the keyframe ID (4 bytes), spatial location of keypoint (4 bytes), scale (2 bytes), dominant orientation (2 bytes), and Hamming signature (4 bytes). Thus, the space is linear to the number of keypoints to be indexed. For our dataset of 144G bytes and 1,040 hours of videos, the size of inverted file is 2.45G.

V. Tag Annotation on Web Videos

Once the near-duplicate or partial near-duplicate videos of a given query are retrieved, a pool of user tags associated with these videos can be acquired. Tags associated with videos are usually in the form of a freely-chosen and short list of keywords. These keywords might give descriptions to video content, with additional context information. Since most users are lazy and not expected to spend much time to tag videos, it is expected the tags will be incomplete, diverse and with redundant and noisy information. Spelling errors and special characters may appear frequently. The mission of data-driven annotation is to select a small number of representative tags to annotate the un-tagged videos.

A scenario we assume here is that tags commonly given by users are more likely related to the actual content of videos, and less likely to be spam. Generally speaking, these tags have higher chance of being re-used for videos of similar content. In other words, tag frequency gives clue to the relevance of a tag to videos. In addition, the number of tags given per video has several implications. On the positive side, a large number of tags hints a diverse video content. On the downside, these tags may simply be a verbose description of a video content when there are few or no tags which can uniquely describe the video content. In contrast, for a video tagged with few keywords, it is reasonable to expect that these keywords are given based on impression or intuition directly observable from the video. Such tags are more likely related to the main theme and content of videos.

Based on this intuition, we propose a measure to rank the relevance of tags based on tag frequency, the number of tags associated to a video, and the video similarity. Let $\Theta_i = V_1, V_2, ..., V_m$ be the set of m near-duplicate videos being retrieved and $\Delta = t_1, t_2, ...$ be the set of tags associated with videos in Θ_i. The relevancy of a tag t_i to query Q is defined as:

$$score(t_i) = \sum_{k=1}^{m} \frac{tf_{ik}}{|V_k|} \times \mathcal{S}im_{re}(V_k, Q), \qquad (18)$$

where tf_{ik} is a binary value of 0 or 1, denoting the absence or presence of tag t_i in V_k, $|V_k|$ is the number of tags with V_k, and $\mathcal{S}im_{re}(V_k, q)$ is the video-level similarity. Notice that the importance of a tag is also reflected by the similarity between Q and V_k. In other words, the tags from videos which are fully duplicate is expected to carry higher weights than those from videos which are partially duplicate to Q. Ultimately, the score of tags determines the rank list of tags recommended to the query Q for annotation. Notice that the ability to rank tags according to their relevancy and popularity can indeed alleviate the adverse effect introduced by noisy user-tags. As a result, the rank list provides a more complete tag list, and in addition, the subjective tags are pushed to the bottom of list.

VI. Experiment Setup

A. Datasets

To verify the robustness, effectiveness and efficiency of the proposed works for web video annotation, two web video datasets (**DS_SOURCE** and **DS_TIME**) are collected to evaluate the performance. Dataset **DS_SOURCE** was collected in November, 2006, which includes videos from YouTube, Google, and Yahoo! video search engines. Our aim is to annotate the Google and Yahoo! videos using YouTube resource. We selected 24 queries designed to retrieve the most viewed and top favorite videos from YouTube. Each text query was issued to YouTube, Google video, and Yahoo! video separately and we collected all retrieved videos as our DS_SOURCE. This collection consists of 12,790 videos, which is the same dataset used in [29], [30]. To test the scalability of near-duplicate detection, we further downloaded another 5,000 videos from YouTube using different sets of queries. The final dataset eventually consists of 1,040 hours of videos. We use 1,428 + 462 videos from Google and Yahoo! respectively as the testing queries because they have no associated tags. The query information and the number of near-duplicates in DS_SOURCE are listed in Table I. The 3rd column shows the number of videos from YouTube, while the number of web videos from Google and Yahoo! is listed in the 4th column. The 5th and 6th columns list the number and percentage of near-duplicates among which videos from Google and Yahoo! can find the corresponding near-duplicates in the YouTube. For example, 85 out of 93 videos from Google and Yahoo! in Query 19 ("Sony Bravia") have corresponding near-duplicate videos in YouTube. On average, there are 38.6% videos from Google and Yahoo! having counterparts in YouTube.

Since the distribution of video data evolves as the time goes by, it is expected that the uploaded videos might deviate from the original ones, and the number of near-duplicate videos should diminish. To verify the robustness of proposed method, dataset **DS_TIME** was collected in December 2008, using the same queries as in DS_SOURCE but crawled at different time. The objective is to annotate the newly added videos using previous data. Table II shows the details. For each topic, we retrieve top 500 videos returned by YouTube search engine. Among these videos, the ones having time overlapping with videos in the first dataset DS_SOURCE are removed. In Table II, the third column lists the number of videos in

TABLE I
THE INFORMATION OF DATASET DS_SOURCE. ANNOTATING VIDEOS OF GOOGLE AND YAHOO!(G+Y) BY USING YOUTUBE (YT)

ID	Topic	YT	G+Y	ND	%
1	The lion sleep tonight	664	128	61	47.7
2	Evolution of dance	354	129	29	22.5
3	Fold shirt	337	99	61	61.6
4	Cat massage	293	51	22	43.1
5	Ok go here it goes again	263	133	17	12.8
6	Urban ninja	682	89	40	44.9
7	Real life Simpsons	249	116	40	34.5
8	Free hugs	489	50	32	64
9	Where the hell is Matt	196	39	9	23.1
10	U2 and green day	277	39	8	40.0
11	Little superstar	229	148	18	12.2
12	Napoleon dynamite	769	112	33	29.5
13	I will survive Jesus	376	40	30	75.0
14	Ronaldinho ping pong	79	28	11	39.3
15	White and Nerdy	1495	276	70	25.4
16	Korean karaoke	180	25	16	64.0
17	Panic at the disco I write sins not tragedies	609	38	9	23.7
18	Bus uncle	453	35	23	65.7
19	Sony Bravia	473	93	85	91.4
20	Changes Tupac	178	16	9	56.3
21	Afternoon delight	412	37	10	0.27
22	Numa Gary	360	62	34	54.8
23	Shakira hips don't lie	1146	176	42	23.9
24	India driving	220	67	13	19.4
-	Others*	5000	-	-	-
	Total	15720	1870	722	38.6%

* Others are videos randomly downloaded in 2009 using queries different from the 24 topics.

TABLE II
THE INFORMATION OF DATASET DS_TIME. ANNOTATING NEWLY CRAWLED DATASET DS_TIME (NEW) BY PREVIOUSLY CRAWLED DS_SOURCE (OLD)

ID	Topic	OLD	NEW	ND	%
1	The lion sleep tonight	792	395	82	20.8
2	Evolution of dance	483	414	9	2.2
3	Fold shirt	436	355	35	9.9
4	Cat massage	344	433	10	2.3
5	Ok go here it goes again	396	255	13	5.1
6	Urban ninja	771	337	53	15.7
7	Real life Simpsons	365	304	20	6.6
8	Free hugs	539	324	0	0.0
9	Where the hell is Matt	235	437	7	1.6
10	U2 and green day	297	328	54	16.5
11	Little superstar	377	397	0	0.0
12	Napoleon dynamite	881	326	47	14.4
13	I will survive Jesus	416	326	240	73.6
14	Ronaldinho ping pong	107	160	61	38.1
15	White and Nerdy	1771	334	76	22.8
16	Korean karaoke	205	350	11	3.1
17	Panic at the disco I write sins not tragedies	647	375	48	12.8
18	Bus uncle	488	250	0	0.0
19	Sony Bravia	566	392	65	16.6
20	Changes Tupac	194	446	75	16.8
21	Afternoon delight	449	426	37	8.7
22	Numa Gary	422	375	89	23.7
23	Shakira hips don't lie	1322	342	92	26.9
24	India driving	287	378	8	2.1
-	Others*	5000	-	-	-
	Total	17790	8459	1141	13.5%

* Others are videos randomly downloaded in 2009 using queries different from the 24 topics.

the DS_SOURCE dataset, while the fourth shows the number of web videos newly collected. The 5th and 6th columns demonstrate the number and percentage of newly crawled videos that can find the near-duplicate in the DS_SOURCE dataset. Query "I will survive Jesus" still has a high percentage of near-duplicates (73.3%) for videos uploaded from December 2006 to December 2008. Based on our statistics, the average percentage of near-duplicate videos is 13.5%, which is smaller than the percentage in DS_SOURCE. Currently a portion of fully duplicate videos has been removed by YouTube, and there is a two year interval between these dataset collections. The topics may become unpopular and the contents can deviate from the original ones. These are the reasons for a relatively lower percentage of near-duplicates compared to DS_SOURCE.

According to our statistics, among the 722 (1,141) near-duplicates used as testing queries in DS_SOURCE (TIME), there are approximately 35.9% (27.2%) having at least one exact duplicate in the reference set. Another 63.1% (72.8%) of queries have near-duplicates due to various forms of editing effects or changes in camera viewpoint. Among them, some videos are either trimmed or inserted with new materials resulting in partial near-duplicates.

B. Pre-Processing

Shot boundaries are detected and each shot is represented by a keyframe. Totally, there are 398,009 keyframes in DS_SOURCE. Local keypoints are detected by Harris-Laplace and described by SIFT [17]. For learning visual dictionary, we collect 742,139 local features from 2,000 keyframes which are randomly selected from the DS_SOURCE dataset. The publicly available toolkit CLUTO [11] is employed to cluster local points into 20,000 clusters. To compare the performance, we also extract the color moment feature. Each keyframe is depicted with the first three color moments (i.e., mean, standard deviation, and skewness) extracted in Lab color space over 5×5 grid partitions, which results in a 225 dimensional feature vector.

Due to the noisy user-supplied tag information, special characters (e.g., ?, !, :, #, $>$, $|$) are first removed. Then the standard Porter stemming is applied to stem the text words. After a serial of data preprocessing (such as word stemming, special character removal, Chinese word segmentation, and so on), there are 14,218 unique tag words.

C. Performance Metrics

1) Near-Duplicate Video Retrieval: To generate the ground-truth, two assessors are asked to watch the query and then browse through the videos in datasets to manually find the corresponding near-duplicate videos. The labeling is based on visual impression that the videos which are transformed versions of one another, showing changes either because of editing operations or camera settings or any combination of them, are regarded as near-duplicates. Partial near-duplicate videos, with at least one shot being near-duplicate, are also included as the ground-truth.

We use *recall*, *precision* and *accuracy* to evaluate the retrieval performance. These measures examine the ability to retrieve all relevant matches (recall), to minimize false positives (precision), and to signal alarm if the query is novel (accuracy). Recall refers to the percentage of near-duplicate

TABLE III
PERFORMANCE COMPARISON OF NEAR-DUPLICATE VIDEO RETRIEVAL ACROSS SOURCES ON DS_SOURCE.

	Precision					Recall				
Topic	LSH-E	VK	VK+	VK++	VK#	LSH-E	VK	VK+	VK++	VK#
1	0.989	0.995	0.915	0.983	**0.999**	0.900	0.917	0.852	**0.986**	**0.986**
2	0.927	0.403	0.269	0.861	**0.987**	0.354	0.594	0.302	**0.969**	0.906
3	0.926	0.992	0.929	0.992	**0.997**	0.803	0.869	0.896	**0.973**	0.940
4	**1.000**	**1.000**	0.999	**1.000**	**1.000**	0.952	0.979	**1.000**	**1.000**	**1.000**
5	0.987	0.847	0.735	0.988	**0.993**	0.861	0.937	0.949	0.962	**0.987**
6	0.044	0.865	0.629	0.995	**0.998**	0.729	0.940	0.977	0.985	**0.992**
7	0.980	0.983	0.824	0.999	**1.000**	0.856	0.949	0.966	**1.000**	**1.000**
8	0.050	0.880	0.683	0.981	**0.994**	0.612	0.633	0.633	**0.796**	0.755
9	0.213	0.136	0.278	0.936	**0.978**	0.923	**0.962**	**0.962**	**0.962**	**0.962**
10	0.113	0.68	0.341	0.874	**0.917**	0.646	0.954	0.969	0.969	**0.985**
11	0.137	0.991	**1.000**	0.982	**1.000**	0.609	0.652	0.841	0.986	**1.000**
12	**1.000**	0.876	0.958	0.995	0.994	0.490	0.903	0.916	0.961	**0.968**
13	0.998	**1.000**	0.999	**1.000**	**1.000**	0.723	0.992	0.863	**1.000**	**1.000**
14	0.222	0.172	0.390	0.969	**0.973**	0.019	0.925	0.925	0.925	**1.000**
15	0.525	0.903	0.865	0.997	**0.999**	0.872	0.964	0.968	**0.974**	0.971
16	0.110	0.887	0.965	**0.993**	**0.993**	0.827	0.865	0.923	0.962	**1.000**
17	0.263	0.937	**0.984**	0.935	0.941	0.310	0.957	0.933	0.995	**1.000**
18	0.060	0.373	0.494	0.707	**0.926**	0.412	0.897	0.779	0.956	**1.000**
19	0.035	0.925	0.875	0.991	**0.995**	0.218	0.979	0.982	**0.994**	0.991
20	0.979	0.539	0.428	0.928	**0.998**	0.625	0.900	0.875	**0.975**	**0.975**
21	0.031	0.684	0.435	0.978	**0.992**	0.382	0.985	**1.000**	**1.000**	**1.000**
22	0.217	0.554	0.264	0.900	**0.977**	0.538	0.670	0.681	**0.901**	**0.901**
23	0.198	0.772	0.817	0.972	**0.984**	0.625	0.701	0.669	0.982	**0.985**
24	0.380	0.151	0.230	0.977	**1.000**	0.651	0.721	0.884	**0.930**	0.884
Average	0.474	0.731	0.679	0.956	**0.985**	0.622	0.869	0.864	0.964	**0.966**

videos being correctly retrieved compared to the ground-truth. Precision means the percentage of correctly retrieved videos among the returned videos. Accuracy refers to the percentage of queries which are correctly judged as having near-duplicates to a target dataset.

2) Tag Annotation: To evaluate the annotation performance, one way is to directly compare the tags generated by the proposed approach with the original ones supplied by users. Nonetheless, those tags tend to be noisy, incomplete and ambiguous. Simply treating user tags as ground truth is not completely objective. Therefore, we adopt manual labeling to generate the ground truth. First, we pool the keywords from tags and titles of near-duplicate videos. Then, for each video, the keywords are recommended, one after another from the pool, to the assessors. The assessors determine whether to accept the keywords as tags, and have option to add new tags after browsing the video and suggest tags from the pool. The evaluation is conducted by comparing the tag set T_S suggested by the proposed approach in Section V and the ground truth T_G. The overlap between T_S and T_G is then used to examine the tag quality. In other words, a tag is regarded correct if the tag also appears in the ground truth. Similar to previous works on image annotation [21], [23], we adopt three measures as the performance metrics, which evaluate the performance from different aspects:

- *Mean Reciprocal Rank (MRR)*: MRR measures the reciprocal of the position in the ranking where the first relevant tag is returned by the system, averaged over all the videos. This measure provides insight on the ability of the system to return a relevant tag at the top of the ranking. The value of MRR is within the range of [0, 1]. A higher score indicates a better performance. A value of 1 indicates that all top one ranked tags are relevant.
- *Success at Rank K (S@K)*: S@K measures the probability of finding a good descriptive tag among the top *k* recommended tags.
- *Precision at Rank K (P@K)*: P@K measures the proportion of retrieved tags that are relevant at rank *k*.

VII. EVALUATION

A. Performance of Near-duplicate Video Retrieval

A critical step in model-free annotation process is to locate the near-duplicate videos for the query video. The performance of near-duplicate video retrieval directly affects the quality of recommended tags. As a result, the effectiveness of retrieval is one of the major concerns.

We compare the performance of following approaches: 1) LSH-E: locality sensitive hashing embedding on color moment, 2) VK: visual keyword search together with inverted file index and 2D HT for video similarity measure, 3) VK+ [7]: visual keyword search with WGC checking and 2D HT, 4) VK++: visual keyword search plus E-WGC checking and 2D HT, and 5) VK#: VK++ with RE ranking. LSH-E is a recently proposed technique in [3] for scalable video search. Using LSH, the technique embeds color moment into a long and sparse feature vector. Inverted file is then applied to support the fast retrieval of long vectors. We use the same implementation provided by [3]. Cosine distance measure is used for forming the embedded space. The corresponding parameters: number of hash functions in a histogram (B=10), folding parameter (M=4), and number of histogram (N=18) are optimized and set according to [3]. The bin width δ_0 in 2D Hough Transform is tested with different settings (from 150 to 240). Due to the space limitation, details will not be presented in this paper. We use the best possible setting of bin width which equals to 200 for all approaches.

TABLE IV
PERFORMANCE COMPARISON OF NEAR-DUPLICATE VIDEO RETRIEVAL ACROSS TIME ON DS_TIME.

	Precision					Recall				
Topic	LSH-E	VK	VK+	VK++	VK#	LSH-E	VK	VK+	VK++	VK#
1	0.934	0.131	0.177	0.996	**0.999**	0.879	0.803	0.797	0.953	**0.962**
2	0.241	0.845	0.403	0.890	**0.991**	0.084	0.864	0.876	0.918	**0.936**
3	0.732	0.936	0.512	0.971	**0.996**	0.751	0.852	0.975	0.911	**0.928**
4	**1.000**	0.996	0.992	0.995	0.997	0.470	0.929	**0.958**	**0.958**	**0.958**
5	0.205	0.343	0.368	0.946	**0.984**	0.092	0.956	0.981	0.955	**0.977**
6	0.070	0.468	0.098	0.887	**0.946**	0.833	0.908	**0.954**	0.937	0.943
7	0.978	0.759	0.620	0.977	**0.995**	0.370	0.851	0.856	0.978	**0.983**
8	-	-	-	-	-	-	-	-	-	-
9	0.002	0.073	0.155	**0.966**	**0.956**	0.059	0.912	**0.971**	0.912	0.912
10	0.554	0.553	0.241	0.975	**0.984**	0.678	**0.920**	0.862	0.828	0.805
11	-	-	-	-	-	-	-	-	-	-
12	0.721	0.689	0.700	0.993	**0.994**	0.404	0.858	**0.896**	**0.896**	0.869
13	0.339	0.937	0.915	**1.000**	**1.000**	0.699	0.985	0.985	0.992	**0.995**
14	0.006	0.142	0.081	0.984	**0.99**	0.075	**1.000**	0.988	0.975	0.950
15	0.732	0.872	0.652	0.977	**0.998**	0.884	0.959	0.979	**0.980**	**0.980**
16	0.058	0.172	0.124	0.941	**0.974**	0.636	0.886	**0.955**	**0.955**	**0.955**
17	0.675	0.429	0.369	0.978	**0.994**	0.341	0.948	0.943	**0.983**	**0.983**
18	-	-	-	-	-	-	-	-	-	-
19	0.158	0.600	0.380	0.987	**0.994**	0.807	0.890	0.976	0.968	**0.971**
20	0.380	0.262	0.078	0.913	**0.986**	0.641	0.946	**0.967**	0.957	0.946
21	0.076	0.677	0.651	**0.999**	**0.999**	0.636	0.795	0.727	0.841	**0.852**
22	0.135	0.131	0.053	0.900	**0.926**	0.426	0.792	0.802	**0.911**	0.901
23	0.251	0.740	0.689	0.992	**0.994**	0.436	0.870	0.949	0.952	**0.957**
24	0.391	0.903	0.743	0.956	**0.966**	**0.960**	0.920	**0.960**	**0.960**	**0.960**
Average	0.411	0.555	0.429	0.963	**0.984**	0.531	0.897	0.922	**0.939**	**0.939**

To decide whether a video should be labeled as near-duplicate, we adopt thresholding technique for all five tested approaches. In the experiment, the best possible thresholds are separately identified for each approach on a subset of DS_SOURCE dataset. The thresholds are then applied to all experiments on DS_SOURCE and DS_TIME datasets.

Table III and IV list the performance of near-duplicate video retrieval in DS_SOURCE and DS_TIME datasets, respectively. The experiments are conducted for queries with near-duplicates. Therefore, 722 and 1,141 queries are tested for DS_SOURCE and DS_TIME, respectively. Overall, VK based approaches show significantly better performance than LSH-E. LSH-E is based on color feature, which lacks discriminative power as the dataset becomes larger. Local feature is a more favorable choice than global features. Visual keyword based search generally returns a high recall of near-duplicate keyframes but mixed with a large portion of false matches. Although geometric consistency constraint WGC is employed by VK+, there is no obvious improvement over VK. On the contrary, E-WGC significantly enhances the performance of VK++ and VK#, especially for precision. Compared to VK+, the improvement of VK++ in terms of precision is over 41% and 124% for DS_SOURCE and DS_TIME, respectively. In addition to scale and rotation transforms, E-WGC also integrates the translation transform, which is a more discriminative measure. Only matches following the same linear transformation will be kept, which filters out large amount of false positives while maintaining a high recall. Hough Transform (HT) weakly considers the temporal consistency. Unfortunately, its effectiveness can be affected by noisy matches. With the assistance of reverse entropy (RE), the performance is further boosted. VK# achieves the best performance. Noisy keyframe matches are effectively pruned out, which reduces the possibility of inducing noises into tag annotation.

TABLE V
MAP PERFORMANCE OF NEAR-DUPLICATE VIDEO RETRIEVAL

Dataset	LSH-E	VK	VK+	VK++	VK#
Cross Source	0.514	0.690	0.684	0.849	**0.869**
Cross Time	0.290	0.463	0.519	0.721	**0.762**

Note that the performances of LSH-E, VK and VK+ fluctuate across 24 topics. For topics containing large number of exact duplicates (or copies), these approaches normally exhibit satisfactory performance. Such topics include “I will survive Jesus” (topic-13). On the other hand, for topics having different versions of near-duplicates as a result of various editing effects, the performances are usually unsatisfactory. One example is “Bus uncle” (topic-18) where the original version is captured by a cell phone. Due to different edited versions as well as the low visual quality of original video, the retrieval results are generally poor. Nevertheless, for VK++ and VK#, due to the consideration of robustness in geometric checking and video-level similarity aggregation, their performances are consistently good for nearly all the tested topics.

To confirm the retrieval performance of five approaches, we also measure the average precision (AP) of each query topic. AP is a measure indicating the area under a precision-recall curve. We measure AP of up to top-500 retrieved items. The mean AP (MAP) of the five approaches is listed in Table V. Overall, VK# and VK++ performs significantly better than other approaches.

B. Performance of Tag Annotation

We evaluate the quality of tags returned for the queries which have near-duplicates in the reference set. Since the

average number of user-supplied tags for each video is around 4.6 in the dataset, we evaluate the performance of the top 5 tags in the experiment. In order to evaluate the effect of the video similarity to tag recommendation, we also test the performance of VK# but ignoring the weight of video similarity in Eqn. 18. That is, the video similarity score is treated as a binary measure. It will contribute 1 if a near-duplicate video is identified, otherwise it is 0. Therefore, the approaches is called VK# (Binary). The performance of cross-source and cross-time annotation is measured according to MRR, S@5 and P@5, which is shown in Table VI. Overall, the performance of near-duplicate video retrieval directly affects the quality of tag annotation. Theoretically, retrieving more near-duplicate videos can supply more information for the tag annotation. In practice, as long as a subset of representative near-duplicate videos with quality tags are retrieved, it already provides the most essential tags for recommendation. As indicated by MRR, the suggested tags at the top ranking are meaningful for all approaches. Therefore, the performance difference among various approaches is not as apparent as the results of near-duplicate retrieval.

Both VK and VK+ show less satisfactory performances across two datasets. Among the retrieved videos, a large number of videos are falsely detected as near-duplicates, which cause the pool of candidate tags to be relatively noisy. On the contrary, with an accurate near-duplicate video retrieval, VK++ and VK# offer comparably better annotation quality. The P@5 measure of VK# is around 0.8, which exhibits satisfactory annotation accuracy. It means that among top 5 recommended tags, nearly 4 of them are closely relevant to the video content.

To compare the results to a baseline (termed as "Oracle" in Table VI), we also show the ideal performance when all ground-truth near-duplicates are perfectly detected. In other words, Oracle is the results with the similarity scores of true near-duplicates being set to 1 (see Eqn 18), while others are weighted with zero scores. From Table V, it can be noticed that the results of VK++ and VK# indeed approach the ideal performance.

The top 5 representative tags of some examples are illustrated in Figure 5. We can see that the suggested tags are meaningful. For example, the second video belongs to "White and Nerdy". In addition to the common words: "white", "nerdy" the suggested tags provide new clues and useful information for this video, such as the author name "al yankovic", and his theme "weird al". It is able to summarize the comments or view points from several users regarding this set of similar videos. In addition, abstract tags such as "funny", which is difficult to train with model-based approaches, is also included for two videos in Figure 5. This tag, although somewhat subjective, is commonly used to annotate similar versions of videos in our dataset, and is also marked as appropriate by the assessors. Content-related tags are also suggested. The third video in Figure 5 is a commercial for "Sony Bravia" TV set. The dominant content in this video is series of balls bouncing, and the content is captured by the tags "bounce" and "ball". Nevertheless, compared to model-based approaches which can offer better recall in providing consistent labeling of prominent visual content, the performance of search-based annotation is dependent on whether the right content of videos is always correctly tagged by users.

TABLE VI
PERFORMANCE COMPARISON FOR TAG ANNOTATION

	Cross Source			Cross Time		
Method	MRR	S@5	P@5	MRR	S@5	P@5
LSH-E	0.801	0.872	0.641	0.677	0.807	0.535
VK	0.860	0.930	0.704	0.831	0.895	0.675
VK+	0.868	0.933	0.714	0.826	0.911	0.674
VK++	0.961	0.973	0.783	0.956	0.983	**0.818**
VK# (Binary)	0.901	0.968	0.716	0.891	0.949	0.763
VK#	**0.962**	**0.985**	**0.785**	**0.959**	**0.988**	0.815
Oracle	0.967	0.991	0.810	0.990	1.000	0.861

Fig. 5. Examples for tag annotation.

C. Beyond Model-free Annotation

The previous two subsections present the performance based on the queries which have near-duplicates in the datasets. For queries which are novel, proper message should be signaled such that no tags will be recommended for annotation. In this subsection, we study the ability of four approaches in determining novel videos. We use all new videos (8,466 in total) in DS_SOURCE and DS_TIME as queries for testing the capability of identifying these new videos. Accuracy is employed as the evaluation measure. The performance of four approaches in DS_SOURCE and DS_TIME datasets is listed in Table VII, in which the accuracy is averaged over 24 topics. VK and VK+ demonstrate poor performance in both datasets. Around 15% and 40% of novel videos falsely identify near-duplicate videos by VK in the reference dataset for DS_SOURCE and DS_TIME, respectively. The performance in DS_TIME is worse than in DS_SOURCE. Since the number of novel videos (7,318) tested in DS_TIME is much larger then the one in DS_SOURCE (1,148), it causes a drop in performance. With E-WGC, VK++ significantly improves the performance, in which a considerable portion of false matches is successfully eliminated for DS_TIME. By incorporating RE, VK# boosts the performance, especially for DS_TIME. Because datasets DS_SOURCE and DS_TIME are

TABLE VII
PERFORMANCE COMPARISON FOR IDENTIFYING NOVEL QUERIES

Method	Cross Source	Cross Time
VK	0.850	0.590
VK+	0.831	0.420
VK++	0.926	0.805
VK#	**0.974**	**0.904**

collected from various sources and at different time periods, the videos in the datasets may demonstrate totally different content. Therefore, E-WGC and RE demonstrate inconsistent improvement for these two datasets. However, we can see that E-WGC and RE complement each other for different datasets. Their combination makes VK# a robust approach for identifying near-duplicate videos while pruning false matches. More than 97% and 90% novel videos can be correctly signaled for two datasets.

D. Speed Efficiency and Scalability

The efficiency is a critical factor for consideration, especially for web scale applications. Mainly the following factors affect the computation cost: keyframe extraction, local point feature extraction, visual keyword quantization, and near-duplicate video retrieval on inverted file. The time costs of these four factors are listed in Table VIII. According to our experiment, time cost for HT, RE and tag recommendation is negligible. Our programs are implemented in C++, and performed on a PC with Intel Duo Core 3.2GHz CPU and 3G memory. Among all approaches, LSH-E is the fastest. For one query video, it takes less than 1.5s to fulfill the retrieval. The time costs for VK based approaches are much higher because they are operated at the keyframe level and using local features.

In our dataset, the average length of a query is approximately 3 minutes with about 53 keyframes, and 260 keypoints per keyframe. To process a query of average length, as shown in Table VIII, in total it takes 84 seconds for VK, and 98.6 seconds for VK+, VK++ and VK#. Without geometric consistency checking, VK is about 15% times faster. From our observation, the use of HE indeed helps to prune significant amount of potential matches that otherwise have to be processed by WGC and E-WGC. Compared to more costly but widely used geometric checking such as RANSAC, WGC and E-WGC are approximately 40 times faster in our experiments.

In our current implementation, the most time consuming parts are the extraction of keypoints and VK quantization. The excessive number of keyframes and keypoints practically makes VK based approaches less scalable if compared to LSH-E. To trade off speed and effectiveness, possible adjustment includes using less keyframes and less keypoints. To provide insights, we experiment a new setting by extracting one keyframe per shot from query and reducing the original amount of keypoints by half per keyframe, which results in an average of 35 keyframes per query. Under this new setting, VK# only requires 33.1 seconds on average to complete a query. We conduct simulation for cross-time tagging. The annotation performance drops slightly (MRR=0.949, S@5=0.978, P@5=0.804), but is still competitive compared to the original results. Detecting near-duplicates also becomes slightly less effective with recall=0.921, precision=0.978 and MAP=0.663.

TABLE VIII
TIME COST COMPARISON

Method	KF Extr.	Feat. Extr.	VK Quant.	Online Retr.
LSH-E	1.0 s	0.4 s	-	0.016 s
VK	1.0 s	1.1×53 s	0.41×53 s	0.082×53 s
VK+	1.0 s	1.1×53 s	0.41×53 s	0.35×53 s
VK++	1.0 s	1.1×53 s	0.41×53 s	0.35×53 s
VK#	1.0 s	1.1×53 s	0.41×53 s	0.35×53 s

The average number of keyframes in one query video is 53.

In practice, for applications which do not require instant query response, VK# indeed offers reasonable trade-off between speed and effectiveness. For instance, in the case of cross-source tagging where the purpose is for improving search performance of an engine, the tagging can indeed be performed offline. For cross-time tagging, if user interaction is not expected, VK# can be run in the background to incrementally enrich or extend the tags provided by users. Otherwise, LSH-E or VK (with lower sampling rate of keyframes and keypoints) could be a better choice, in which users can simply and instantly select appropriate tags recommended by machine, though with loss in recall for all relevant tags.

VIII. CONCLUSION AND DISCUSSION

With the rival growth of social media, there are abundant video resources available online. They are usually accompanied by user-supplied tags, which provide a valuable resource to explore. Different from traditional approaches which adopt computer vision or machine learning techniques, we investigate the potential of a data-driven and model-free approach to annotate the web videos. A simple but effective solution is proposed by employing the near-duplicate video retrieval techniques and classifier-free video annotation. An effective near-duplicate retrieval approach which integrates enhanced weak geometric constraint (E-WGC), Hough Transform (HT), and reverse entropy (RE) has been proposed. In particular, VK# has shown superior capability in pruning out false alarms compared to the-state-of-art VK technique. Representative tags are then recommended to annotate using videos from different sources or time. Experiments on cross-source and cross-time datasets demonstrate the effectiveness and robustness of the proposed data-driven approach. Currently, the proposed VK based approaches cannot be directly extended to cope with global features. Further research is required to investigate the proper integration of global and local features for more reliable and speedy search-based annotation.

Our approach annotates videos by exploiting visual duplicates. As indicated in our cross-source and cross-time datasets, there are 38.6% and 13.5% videos having corresponding visual duplicates. In other words, a larger portion of novel videos are uploaded each day. For novel videos which cannot find the counterpart in the reference dataset, our approach is unable to provide meaningful tags. However, once the videos could find at least one near-duplicate and as the size of reference dataset continues to grow, our approach can take effect. As a

remark, the works presented in this paper indeed provide an efficient way of annotating videos which "look familiar" or are "partially new but previously seen".

REFERENCES

[1] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *SIGCHI*, pages 971–980, 2007.

[2] P.-A. Chirita, S. Costache, S. Handschuh, and W. Nejdl. P-tag: Large scale automatic generation of personalized annotation tags for the web. In *Proc. World Wide Web*, pages 845–854, 2007.

[3] W. Dong, Z. Wang, M. Charikar, and K. Li. Efficiently matching sets of features with random histograms. In *Proc. ACM Conf. Multimedia*, pages 179–188, 2008.

[4] M. Douze, A. Gaidon, H. Jegou, M. Marszatke, and C. Schmid. INRIA-LEAR's video copy detection system. In *TREVCID*, 2008.

[5] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[6] A. Hampapur and R. Bolle. Comparison of sequence matching techniques for video copy detection. In *Proc. Conf. on Storage and Retrieval for Media Databases*, pages 194–201, 2002.

[7] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. European Conf. on Computer Vision*, pages 304–317, Oct. 2008.

[8] Y.-G. Jiang and C.-W. Ngo. Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval. *Computer Vision and Image Understanding*, 113:405–414, 2009.

[9] A. Joly and O. Buisson. A posteriori multi-probe locality sensitive hashing. In *Proc. ACM Conf. Multimedia*, pages 209–218, 2008.

[10] A. Joly, O. Buisson, and C. Frelicot. Content-based copy detection using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*, 9(2):293–306, 2007.

[11] G. Karypis. CLUTO. In *http://glaros.dtc.umn.edu/gkhome/views/cluto/*.

[12] Y. Ke, R. Suthankar, and L. Huston. Efficient near-duplicate detection and sub-image retrieval. In *Proc. ACM Conf. Multimedia*, pages 869–876, 2004.

[13] J. Law-To, B. Olivier, V. Gouet-Brunet, and B. Nozha. Robust voting algorithm based on labels of behavior for video copy detection. In *Proc. ACM Conf. Multimedia*, pages 835–844, 2006.

[14] X. Li, L.Guo, and Y. Zhao. Tag-based social interest discovery. In *Proc. World Wide Web*, pages 675–684, 2008.

[15] X.-R. Li, L. Chen, L. Zhang, F.-Z Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *Proc. ACM Conf. Multimedia*, pages 607–610, 2006.

[16] X.-R. Li, C. G. M. Snoek, and M. Worring. Learning tag relevance by neighbor voting for social image retrieval. In *Proc. ACM Conf. Multimedia, MIR workshop*, pages 180–187, 2008.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[18] K. Mikoljczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.

[19] E. Moxley, T. Mei, and B. S. Manjunath. Video annotation through search and graph reinforcement mining. *IEEE. Trans. on Multimedia*, 2009.

[20] G.-J. Qi, X.-S. Hua, and et al. Correlative multi-label video annotation. In *Proc. ACM Conf. Multimedia*, pages 17–26, 2007.

[21] B. Sigurbjornsson and R. Zwol. Flickr tag recommendation based on collective knowledge. In *Proc. World Wide Web*, pages 327–336, 2008.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. Intl. Conf. on Computer Vision*, volume 2, pages 1470–1477, 2003.

[23] Y. Song, Z. Zhuang, H. Li, and et al. Real-time automatic tag recommendation. In *Proc. SIGIR*, pages 515–522, 2008.

[24] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, November 2008.

[25] C. Wang, F. Jing, L. Zhang, and H. J. Zhang. Image annotation refinement using random walk with restarts. In *Proc. ACM Conf. Multimedia*, pages 647–650, 2006.

[26] X.-J. Wang, L. Zhang, X.-R. Li, and W.-Y. Ma. Annotating images by mining image search results. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 30(11):1919–1932, November 2008.

[27] K. Weinberger, M. Slaney, and R. Zwol. Resolving tag ambiguity. In *Proc. ACM Conf. Multimedia*, pages 111–229, 2008.

[28] L. Wu, L. Yang, N. Yu, and X.-S. Hua. Learning to tag. In *Proc. World Wide Web*, pages 361–370, 2009.

[29] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. In *Proc. ACM Conf. Multimedia*, pages 218–227, 2007.

[30] X. Wu, C.-W. Ngo, A. G. Hauptmann, and H.-K. Tan. Real-time near-duplicate elimination for web video search with content and context. *IEEE. Trans. on Multimedia*, 11(2):196–207, Feburary 2009.

[31] L. Wang Y. Jin, L. Khan and M. Awad. Image annotations by combining multiple evidence & wordnet. In *Proc. ACM Conf. Multimedia*, pages 706–715, 2005.

[32] J. Yuan, L.-Y. Duan, Q. Tian, S. Ranganath, and C. Xu. Fast and robust short video clip search for copy detection. In *Pacific Rim Conf. on Multimedia*, pages 479–488, 2004.

[33] D.-Q. Zhang and S.-F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *Proc. ACM Conf. Multimedia*, pages 877–884, 2004.

[34] W.-L. Zhao and C.-W. Ngo. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. *IEEE. Trans. on Image Processing*, 18(2):412–23, Feb. 2009.

PLACE PHOTO HERE

Wan-Lei Zhao received his M.S. and B.C. degrees in Department of Computer Science and Engineering from Yunnan University in 2006 and 2002 respectively. He was with Software Institute, Chinese Academy of Science from Oct.2003 to Oct.2004 as an exchange student. He is currently a Ph.D candidate in Department of Computer Science, City University of Hong Kong. His research interests include multimedia information retrieval and video processing.

PLACE PHOTO HERE

Xiao Wu (S'05, M'08) received the B.Eng. and M.S. degrees in computer science from Yunnan University, Yunnan, China, and Ph.D. degree in the Department of Computer Science from City University of Hong Kong in 2008. His research interests include multimedia information retrieval, video computing, and data mining.

Currently, he is an Associate Professor at Southwest Jiaotong University, Chengdu, China. He was a Senior Research Associate and a Research Assistant at the City University of Hong Kong from 2007 to 2009, and 2003 to 2004, respectively. From 2006 to 2007, he was with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, as a Visiting Scholar. He was with Institute of Software, Chinese Academy of Sciences, Beijing, China, from 2001 to 2002.

PLACE PHOTO HERE

Chong-Wah Ngo (M'02) received his Ph.D in Computer Science from the Hong Kong University of Science & Technology in 2000. He received his MSc and BSc, both in Computer Engineering, from Nanyang Technological University of Singapore. Before joining City University of Hong Kong in 2002, he was with Beckman Institute of University of Illinois in Urbana-Champaign. He was also a visiting researcher of Microsoft Research Asia in 2002. His research interests include video computing and multimedia information retrieval.