Real-time Large Scale Near-duplicate Web Video Retrieval*

Lifeng Shang[†‡], Linjun Yang[‡], Fei Wang[§], Kwok-Ping Chan[†], Xian-Sheng Hua[‡]

[†] The University of Hong Kong, Pokfulam, Hong Kong
[‡] Microsoft Research Asia, Beijing 100190, P. R. China
[§] Microsoft Search Technology Center, Beijing 100190, P. R. China
{lfshang, kpchan}@cs.hku.edu, {linjuny, feiw, xshua}@microsoft.com

ABSTRACT

Near-duplicate video retrieval is becoming more and more important with the exponential growth of the Web. Though various approaches have been proposed to address this problem, they are mainly focusing on the retrieval accuracy while infeasible to query on Web scale video database in real time. This paper proposes a novel method to address the efficiency and scalability issues for near-duplicate Web video retrieval. We introduce a compact spatiotemporal feature to represent videos and construct an efficient data structure to index the feature to achieve real-time retrieving performance. This novel feature leverages relative gray-level intensity distribution within a frame and temporal structure of videos along frame sequence. The new index structure is proposed based on inverted file to allow for fast histogram intersection computation between videos. To demonstrate the effectiveness and efficiency of the proposed method, we evaluate its performance on an open Web video data set containing about 10K videos and compare it with four existing methods in terms of precision and time complexity. We also test our method on a data set containing about 50K videos and 11M key-frames. It takes on average 17ms to execute a query against the whole 50K Web video data set.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process, Selection process, Information filtering*

General Terms

Algorithms, Experimentation, Performance

Keywords

Near-duplicate, Web Videos, Binary Spatiotemporal Feature, Modified Inverted File

*This work was performed when Lifeng Shang was visiting Microsoft Research Asia as a research intern.

MM'10, October 25–29, 2010, Firenze, Italy.

1. INTRODUCTION

With the exponential growth of video sharing websites, e.g. YouTube, the number of videos being searchable on the Web has tremendously increased. Analysis on leading social video sharing platforms reveals a high amount of redundancy with overlapping or duplicate content [20]. Wu et al. [28] showed that there are on average 27% duplicates among the search results of 24 popular queries from YouTube, Google Video and Yahoo Video. To avoid being overwhelmed with duplicates in the video search result and utilize such redundancy for other tasks such as mining the internal structure of video database [20], automatic video tagging [21], etc., it is essential to developing a near-duplicate Web video retrieval system. Unfortunately, most existing near-duplicate video detection systems pay more attention to handle various photometric or geometric transformations [14] [11], which are difficult to handle the Web scale video database and return the search results in real time. Only little work has focused on the real-time response for large scale Web video retrieval [30]. The objective of this paper is to address the efficiency and scalability issues of near-duplicate video retrieval in the Web scale environment, and our method is not for localization task.

The commonly used techniques for near-duplicate video detection system can be summarized as follows [12]. First, videos are segmented into shots and each shot is represented by one or more key-frames. Second, a set of high dimensional feature vectors are extracted to represent the key-frames. The bag of key-frame features is taken as the representation of the whole video. Finally, the similarity between videos is computed by matching key-frame features under spatial and temporal constraints. Based on the described procedure, three important factors that affect near-duplicate detection performance can be identified: key-frame extraction, features representation, and key-frames matching under spatiotemporal constraints. In this paper we aim at solving two of the problems, namely compact video representation and efficient key-frames matching.

Existing work on near-duplicate video identification can be grouped into two categories: global feature based methods and local feature based methods. Global feature based methods extract frame-level signatures to model the information distributed in spatial, color and temporal dimensions [11]. The similarity between query video and database video is calculated as the matching score of the signature sequences. Due to temporal editing, such as inserting in or cutting out short clips, dynamic programming method is often employed for locating a short query video clip in a long target video [12]. Since the state-of-the-art approximate sequence matching algorithm is still very time-consuming, the existing global feature based methods cannot be directly applied for large scale Web video retrieval.

Local feature based methods [7] [14] are very suitable for copyright protection and TV commercial detection, since they can handle various challenging distortions (strong cropping, insertions of large advertisements, picture-in-picture, etc.). However, the time and storage complexity is unacceptably high for a practical real-time application. For example, the state-of-the-art local feature based video copy detection system [7] represents each frame by more than 400 local descriptors, which results in the high computational cost on key-frames matching. Moreover, [7] reported their local feature based system can index at most 2M key-frames, which limits the number of videos to be processed. The high time and storage complexity of local feature based methods lies in hundreds to thousands of local features being extracted for each key-frame, and the number of extracted key-frames can easily exceed 100 even for a short 4 minutes video with fast changing scenes [29].

Compared to global feature based methods, local feature based methods can obtain higher precision, however it requires more time and storage costs. To improve efficiency local feature based methods often extract fewer frames to limit the number of local features being extracted, while this heavily breaks the temporal continuity of videos. The temporal structure of videos plays an important role in near-duplicate video retrieval, while existing temporal local features usually model temporal information by some expensive tracking techniques, which further increases the complexity of local features. In practice only a small portion of near-duplicate Web videos show significant variations, so some simple global features can already achieve satisfactory performance, which was first pointed out by Zhao et al. [30]. Similar observation was also reported by Wu et al. [28] based on extensive experiments on Web video dataset [5]. Therefore, we will mainly focus on global features and particularly study how to model the spatial and temporal information of videos simultaneously in a more compact and robust way. Moreover, we will study how to index the extracted spatiotemporal features and how to efficiently search near-duplicate videos.

This paper has two main contributions, a compact spatiotemporal feature and an efficient near-duplicate video retrieval method which is developed based on the fast intersection kernel [2] and a modified inverted file. Our new spatiotemporal feature models the spatial information of videos by measuring the relative gray-level intensity distribution within a frame. The temporal structure of videos is modeled by the w-$shingling$ [3], which is originally used in text retrieval to measure document similarity. In Section 3, two different strategies are used to describe the spatial information. The first one is automatic and based on conditional entropy (CE) [8]. The second one is heuristic and motivated by the well-known texture descriptor local binary pattern (LBP) [31]. The efficient retrieval method is developed by incorporating the recently proposed fast intersection kernel method [18] [25] into the inverted file. Our framework which combines the compact spatiotemporal feature and the efficient retrieval method provides an effective and fast way to find the near-duplicate videos from a large scale dataset. Experiments on two datasets (one is provided by CityU and CMU and the other is constructed by us) show our method achieves both high precision and real-time performance.

The rest of this paper is organized as follows. Section 2 reviews some related work. Section 3 introduces the proposed spatiotemporal feature. In Section 4, we present the efficient retrieval method. In Section 5, we evaluate the performance of our method on two Web video datasets and compare it to some existing methods. Section 6 summarizes this paper.

2. RELATED WORK

Among existing approaches much attention focuses on global feature based methods, which extract the color, spatial and temporal signature of videos and rely on sequence matching techniques. Color signature encodes the color information of video frames and discards spatial information. Hampapur et al. [11] described a motion signature, that captures the relative change in the intensities over time. Hua et al. [12] employed ordinal signature to model the relative distribution of intensities in a frame. They measured the distance between two video clips by the temporal shape of signatures. Hampapur et al. [11] compared the motion, color and ordinal signatures in the context of copy detection. Experimental results indicated the ordinal signature has superior performance over the other two signatures. Wu et al. [27] introduced a shot boundary based signature and used the suffix array data structure to match signature sequences. Recently, some newly developed techniques (e.g. locality-sensitive hashing [9]) are incorporated into near-duplicate video search. Dong et al. [6] employed LSH to map the color histogram of each key frame into a binary vector. Then the set of binary features are embedded into their proposed histogram representations. Experimental results confirmed the efficiency of this method, while which suffers from the potential problem of huge memory consumption [30].

Extracting local features to represent multimedia data for content based image and video search has been gaining more attention [14] [22]. Douze et al. [7] derived their system from the state-of-the-art content-based image search [13]. The system represented key-frames by sets of SIFT features [17]. Then these local features were quantized into visual words and mapped into binary signatures by Hamming Embedding [13]. Temporal and geometrical parameters were estimated separately with Hough Transforms (HTs). This system can be adapted to handle various difficult transformations (e.g. picture-in-picture). However, it can at most index 2M key-frames and HTs is time-consuming. Thus it cannot be applied to near-duplicate Web video search. The method presented in [15] provided a way to consider the temporal behaviors of local features. It tracked the trajectories of interest points and assigned high level descriptions to each interest point according to trajectory parameters. This method took about 5min for a 24min query video in a 300h database videos [15], so it is not suitable for real-time large scale Web video retrieval. For local feature based methods, to capture the spatial distribution of interest points, matching sets of high-dimensional features has to be conducted, which usually imposes much higher complexity in terms of time and storage. Although some effective matching methods (Pyramid matching [10], RANSAC-based fast image matching [19] and Geometric min-Hashing [4], etc.) have been developed, it is challenging to both maintain the matching quality and reduce the complexity of matching to satisfy real-time requirement.

To meet the speed requirement for large scale Web video collections, Wu et al. [28] proposed a hierarchical method to combine global and local features. They first used color histograms to detect near-duplicate videos with high confidence and filtered out these dissimilar videos. Then a local feature based method was utilized to identify the remainder uncertainty videos. In this method, global feature based method acted as a filter, through which the number of key frames to be matched is reduced in the following more expensive local feature based method. However, this hierarchical method is infeasible for real-time large scale near-duplicate detection, since it still involves a large number of comparisons between key-frames. In [29], Wu et al. fused the contextual information (time duration, thumbnail image, view times, etc.) with video content, which achieved 164 times faster than their hierarchical model with a loss of performance. The strategies introduced in this work to reduce the computation complexity of local feature

based method via incorporating textural information can be directly transferred to other methods. In this work, we will focus on content based near-duplicate video retrieval and other auxiliary information (e.g. contextual or audio) will not be used.

3. BINARY SPATIOTEMPORAL FEATURE

To represent a video, we need to first extract frames from video streams. The commonly used extracting methods are uniform sampling and shot-based method. In uniform sampling method, a fixed number of frames per time unit is extracted, e.g. extracting one frame per second. In shot-based method, a video is segmented into shots and each shot is represented by one or more frames. This method produces a small set of frames (on average one frame every 6 seconds [7]), however it breaks the temporal continuity of videos. In this work, we adopt the uniform sampling to preserve reliable temporal information.

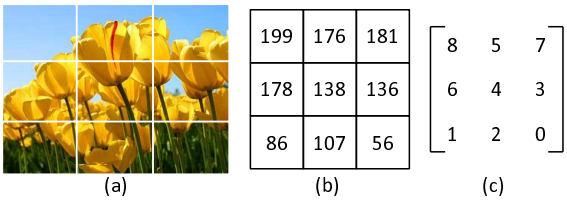

Figure 1: (a) Original image divided into 3×3 blocks. (b) Average gray-level values. (c) Ranks of blocks based on intensity ordering.

The proposed spatiotemporal feature is inspired by ordinal measure, which was first proposed in [1] as a robust feature in image correspondence. Figure 1 depicts the procedure of calculating ordinal measure. Original image is partitioned into 3×3 blocks and the average gray-level value in each block is computed. The set of average intensities is sorted in ascending order and the rank is assigned to each block. Ordinal measure reflects the inherent relative intensity distribution within a single frame, thus it is naturally robust to the color degradation effect caused by different encoding devices [12]. However, this feature does not consider the temporal structure of videos.

Ordinal measure describes the pairwise ordinal relations between blocks in terms of average gray-level values. If a frame was divided into nine blocks, there will be totally 36 (i.e. C_9^2) ordinal relations among the nine blocks. Then the ordinal measure can be rewritten in the form a 36-dimensional binary feature vector (i.e. $\{0,1\}^{36}$). We use set $A_a = \{A_a^1, \cdots, A_a^{36}\}$ to denote the 36 ordinal relations, here A_a^i is the i-th ordinal relation. To get a more compact and robust frame-level representation, we implement feature selection on the 36 ordinal relations. Let $A_a^S = \{A_a^{s(1)}, \cdots, A_a^{s(K)}\}$ denote the K extracted features during the feature selection process. Although many feature selection algorithms have been developed for various machine learning models (e.g. classification) and applications (e.g. face recognition), to our best knowledge how to select appropriate features for large scale near-duplicate Web video retrieval remains untouched. In this section we will present two different methods to extract some ordinal relations from A_a. The first automatic method aims to preserve as much information as possible without considering background knowledge. This method uses the conditional entropy to measure the information loss and the resulted feature is called CE-based spatiotemporal feature. The second heuristic method is motivated by the well-known feature descriptor LBP and developed based on some observations on near-duplicate videos. In the remainder of this paper, we use "STF_CE" and "STF_LBP" to denote the CE- and LBP-based spatiotemporal features, respectively.

3.1 CE-based Spatiotemporal Feature

Our automatic ordinal relations selection method is based on conditional entropy, so we firstly introduce some basic quantities in information theory. The most important quantity of information is entropy, i.e. the information in a random variable. Given a discrete random variable X and it consists of several events x, which occurs with probability $p(x)$, the entropy of X is defined as

$$H(X) = -\sum_x p(x) \log p(x) \tag{1}$$

which is a measure of the amount of uncertainty associated with the value of X. The joint entropy of two discrete random variables X and Y is merely the entropy of their pairing (X, Y) and defined as

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y) \tag{2}$$

where $p(x,y)$ is the joint probability of random variables X and Y. Then the conditional entropy can be easily obtained

$$H(X|Y) = H(X,Y) - H(Y) \tag{3}$$

which quantifies the remaining uncertainty of the random variable X given that the value of Y is known.

The main goal of feature selection is to select a subset of features that carries as much information as possible. From the view point of information theory, the goal of feature selection is to minimize the conditional entropy $H(A_a|A_a^S)$, which quantifies the remaining uncertainty of A_a after giving selected ordinal relations [8]. If the selected ordinal relations contains all the information embedded in A_a, then this conditional entropy is zero, since no more information is required to describe A_a when the selected ordinal relations are known. Estimating the conditional entropy is often time-consuming, since it requires the estimation of 2^K discrete probabilities of a large set of samples, e.g. the number of extracted frames of the dataset [5] is more than 2M. Furthermore, the number of possible choices of the selected ordinal relation set is very large. For example, if the value of K is 12, there will be $C_{36}^{12} \approx 1.25 \times 10^9$ possible choices. Thus, the minimization would be computationally intractable. In this work, we present an efficient ordinal relation selection method that ensures the selected ordinal relations which are both individually informative and two-by-two weakly dependant. Our selection criterion is similar to that presented by Fleuret et al. [8]. The difference is our method is proposed in the context of retrieval rather than classification.

The procedure of ordinal relation selection is summarized in Algorithm 1. The binary matrix $F \in \mathbb{R}^{N \times 36}$ records the ordinal relations for each extracted frame, where N is the total number of extracted frames for a video database and $F_{i,j} \in \{0,1\}$ represents the value of feature A_a^j for the i-th frame. A_a^S and A_a^C are used to represent the selected ordinal relation set and candidate set, respectively. $A_a^{s(i)}$ denotes the i-th selected ordinal relation. $A_a^{c(i)}$ denotes the i-th candidate ordinal relation. In the first step of our algorithm, we initialize A_a^S as the most informative ordinal relation, which is $H(A_a^{s(1)}) \geq H(A_a^i)$ for $1 \leq i \leq 36$. From the definition of entropy, the entropy of the i-th feature A_a^i is calculated by

$$\begin{aligned} H(A_a^i) = &-p(A_a^i = 1) \log p(A_a^i = 1) \\ &-p(A_a^i = 0) \log p(A_a^i = 0) \end{aligned} \tag{4}$$

Algorithm 1 Informative Ordinal Relations Selection

Require: The binary $N \times 36$ matrix F, and the number of features to be selected K.

Ensure: The K individually informative and two-by-two weakly dependent features $A_a^{s(1)}, A_a^{s(2)}, \cdots, A_a^{s(K)}$

(Step 1) Initialize the set A_a^S and A_a^C.
$s(1) = \arg\max_n H(A_a^n)$
$A_a^S = \{A_a^{s(1)}\}$
$A_a^C \leftarrow A_a - A_a^S$

(Step 2) Iteratively select ordinal relations
for k = 1 to $(K-1)$ **do**
$s(k+1) = \arg\max_n \{\min_{l \le k} H(A_a^n | A_a^{s(l)})\}, A_a^n \in A_a^C$
$A_a^S \leftarrow A_a^S \cup \{A_a^{s(k+1)}\}$
$A_a^C \leftarrow A_a - A_a^S$
end for

where $p(A_a^i = 1)$ is the probability of the value of feature A_a^i to be one and is approximated by the relative frequency

$$p(A_a^i = 1) = \frac{\#(\text{frames with the } i\text{-th feature having value 1})}{N} \quad (5)$$

and $p(A_a^i = 0) = 1 - p(A_a^i = 1)$ under the assumption that the N frames are independent. This assumption seems to be restrictive, however our method uses bag-of-words model and the probabilistic dependence of neighboring frames is not considered, so the assumption is satisfied.

In Step 2, ordinal relations are iteratively selected. The $(k+1)$-th ordinal relation is selected as

$$s(k+1) = \arg\max_n \underbrace{\{\min_{l \le k} H(A_a^n | A_a^{s(l)})\}}_{v(n,k;A_a,A_a^S)}, A_a^n \in A_a^C \quad (6)$$

where $H(A_a^n | A_a^{s(l)})$ is the entropy of A_a^n conditional on $A_a^{s(l)}$. This conditional entropy is low if either feature A_a^n is not informative or if its information was already caught by $A_a^{s(l)}$. For a given candidate ordinal relation A_a^n, the smaller the value of $v(n, k; A_a, A_a^S)$ the more information is shared between A_a^n and at leat one of the already selected ordinal relations. Maximizing $v(n, k; A_a, A_a^S)$ ensures the newly selected feature is both informative and weakly dependent on the preceding ones. The calculation of $H(A_a^n | A_a^{s(1)})$ only involves the joint distribution of the two binary variables A_a^n and $A_a^{s(1)}$. From the whole procedure, we can find the maximal number of joint distributions to be calculated is 630 (i.e. $36 \times 35 \times 0.5$), thus the implementation of our ordinal relation selection method can be implemented in a very efficient way.

Table 1 lists the top-18 informative ordinal relations picked up by the fast ordinal relation selection method. If we use the top-8 informative features to represent a frame, then the mapping functions for translating a frame into 8-dimensional binary feature is

$$\mathcal{F}_{info} = \{I(G_{[1,2]} > G_{[2,2]}), I(G_{[1,1]} > G_{[1,3]}), \cdots, I(G_{[1,3]} > G_{[3,1]}), I(G_{[1,2]} > G_{[1,3]})\}, \quad (7)$$

where function $I(S)$ is the indicator function giving 1 if statement S is true and 0 otherwise, $G_{[i,j]}$ is the average gray-level intensity value of the (i,j)-th block ($1 \le i, j \le 3$). In the experimental part, we will set the parameter K empirically. After getting the frame level representation, the temporal structure of videos can be modeled.

Table 1: Top-18 informative ordinal relations

Order	Ordinal Relations	Order	Ordinal Relations
1	$I(G_{[1,2]} > G_{[2,2]})$	10	$I(G_{[1,2]} > G_{[2,1]})$
2	$I(G_{[1,1]} > G_{[1,3]})$	11	$I(G_{[3,1]} > G_{[3,3]})$
3	$I(G_{[3,2]} > G_{[3,3]})$	12	$I(G_{[1,2]} > G_{[2,3]})$
4	$I(G_{[2,1]} > G_{[3,1]})$	13	$I(G_{[1,1]} > G_{[2,1]})$
5	$I(G_{[3,1]} > G_{[3,2]})$	14	$I(G_{[2,1]} > G_{[2,3]})$
6	$I(G_{[1,1]} > G_{[3,3]})$	15	$I(G_{[2,2]} > G_{[2,3]})$
7	$I(G_{[1,3]} > G_{[3,1]})$	16	$I(G_{[2,1]} > G_{[3,2]})$
8	$I(G_{[1,2]} > G_{[1,3]})$	17	$I(G_{[2,3]} > G_{[3,2]})$
9	$I(G_{[2,3]} > G_{[3,3]})$	18	$I(G_{[1,3]} > G_{[2,3]})$

We use the w-$shingling$ concept of text retrieval to model the temporal information [3]. A w-$shingling$ is a set of unique "shingles", i.e. contiguous subsequences of tokens in a document, that can be used to gauge the similarity of two documents. For example, the document "a rose is a rose is a rose" can be tokenized as (a, rose, is, a, rose, is, a, rose). The set of all unique contiguous sequences of 4 tokens is {(a, rose, is, a), (rose, is, a, rose), (is, a, rose, is)} which forms a 4-shingling. Our spatiotemporal signature is very similar to word shingle, except each token is not a word but K-dimensional binary pattern. In our implementation, w is set to 2. Consequently, the binary pattern of video is decomposed into a set of visual shingles. The histogram of visual shingles is used as the bag of visual shingle (BoVS) representation for the original video. The size of visual shingle vocabulary is $2^{2\times K}$.

In traditional "bag-of-words" model, high-dimensional feature vectors are usually quantized into a fixed number of visual words with clustering algorithms (e.g. k-means algorithm). However, building the visual codebooks for millions of high-dimensional feature vectors is very time-consuming. In our method, the mapping functions $\mathcal{F}_{info}$ act as the hash functions to transform frames into binary numbers and encode the spatiotemporal information. Compared to traditional vector quantization, our method can quickly quantize videos into BoVS representations, since there is no need to calculate the distances between a feature vector and all the visual words as traditional quantization method does. A one-time scan of the binary representation can accomplish quantization. Instead of storing the BoVS representations, the compact binary representations are saved to reduce storage cost. In the experimental part, we will try various parameter settings for K to investigate whether preserving more information is sufficient to achieve good performance.

3.2 LBP-based Spatiotemporal Feature

In this section, we will describe an experimentally confirmed heuristic method which is motivated by the well-known feature descriptors LBP and some observations on near-duplicate videos, e.g. the central area of frame is more reliable and repeatable for near-duplicate Web video retrieval. Same to the design of STF_CE, some ordinal relations are selected based on LBP and corresponding mapping functions are defined to translate ordinal measure into a binary representation. The whole procedure of defining STF_LBP is graphically shown in Figure 2. The nine blocks are first separated into two regions, the central region (the shaded region of Fig. 2(b)) and the marginal region (the shaded region of Fig. 2(c)). The four marginal blocks are easily corrupted by Logos (e.g. Logos added by TVs or video sharing platforms). The central region preserves the main content of a video and Logos will generally not be added to this region. Ordinal relations will individually be extracted for the central and marginal regions.

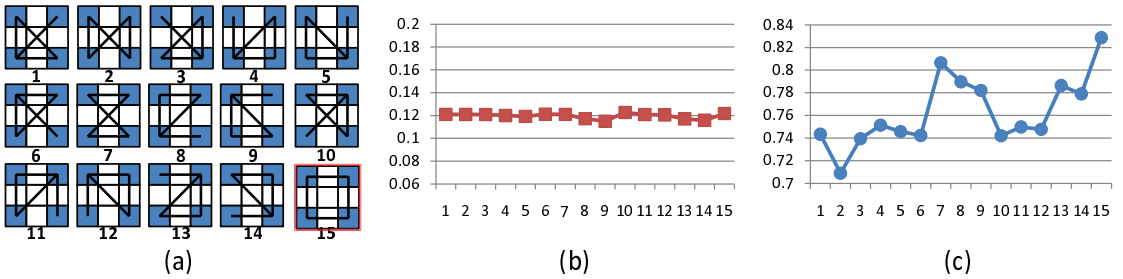

Figure 3: (a) The choices $A_{m,i}$ $(1 \leq i \leq 15)$ and their (b) $U(\bar{A}_c, A_{m,i})$ values and (c) $H(A_{m,i})/H(A_m)$ values

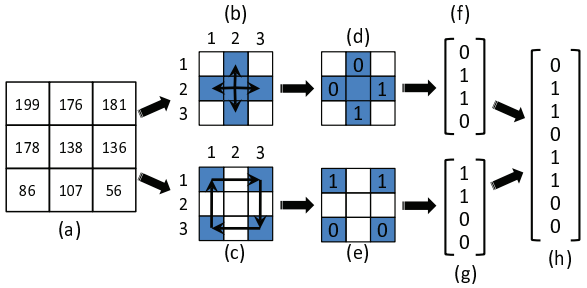

Figure 2: (a) Average gray-level values. (b) Central mapping functions and (c) Marginal mapping functions, the left and top side integers are the row and column numbers, respectively. (d) Mapping results of central functions. (e) Mapping results of marginal functions. (f) Central feature and (g) Marginal feature. (h) Binary frame-level feature.

The selection of ordinal relations for the central region is motivated by the success of LBP, which has been widely used in computer vision (e.g. face detection and facial expression recognition). LBP has the merits of tolerance against illumination changes and computational simplicity, which makes it possible to extract features in challenging real-time settings. Based on the definition of LBP, the mapping functions for the central region are given as

$$\mathcal{F}_c = \{I(G_{[2,2]} > G_{[1,2]}), I(G_{[2,2]} > G_{[2,3]}), \\ I(G_{[2,2]} > G_{[3,2]}), I(G_{[2,2]} > G_{[2,1]})\}. \quad (8)$$

The mapping results of the central functions $\mathcal{F}_c$ is just the state-of-the-art LBP of the center block with neighborhood size four. Fig. 2(b) graphically shows the central mapping functions. Black solid arrow denotes the ordinal relation between the begin block and the end block is selected. We use $\bar{A}_c$ to denote the four selected ordinal relations. Fig. 2(d) shows the mapping results. Fig. 2(f) rewrite the mapping results in vector form, and we call it "central feature".

Based on the primary ordinal relations of the central region, another four auxiliary ordinal relations are selected from the marginal region, which should be informative and have little inter-correlation to the central LBP feature. The correlation between two attributes X and Y is measured by the symmetric uncertainty [24] defined as

$$U(X,Y) = 2\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}, \quad (9)$$

where $H(X,Y)$ is the joint entropy of X and Y, which is calculated from the joint probabilities of all combinations of values X and Y. The symmetric uncertainty lies between 0 and 1. The larger the value of $U(X,Y)$ the more correlation exists between attributes X and Y, and vice versa. Let A_m denote the ordinal relations among the four marginal blocks. It is clear that there are totally 15 possible choices to select 4 ordinal relations from A_m, which are graphically shown in Figure 3(a). In each subfigure, shaded region represents marginal region and the black solid line represents the ordinal relation between the two linked blocks is used. Notation $A_{m,i}$ is used to denote the i-th choice. Based on the open Web video dataset [5], the values of $U(\bar{A}_c, A_{m,i})$ and $H(A_{m,i})/H(A_m)$ are calculated for the 15 choices and plotted in Fig. 3(b) and (c), respectively. The optimal choice should both has the smallest value of $U(\bar{A}_c, A_{m,i})$ and the largest value of $H(A_{m,i})/H(A_m)$. We can observe that $U(\bar{A}_c, A_{m,i})$ are very small values with mean 0.12 and tiny variance 0.53×10^{-5}, which implies all the 15 choices have little inter-correlation to $\bar{A}_c$. Therefore, we adopt the choice which has the largest $H(A_{m,i})/H(A_m)$ value, i.e. $A_{m,15}$. Fig. 2(c) shows the selected ordinal relations from the marginal region and the corresponding mapping functions are

$$\mathcal{F}_m = \{I(G_{[1,1]} > G_{[1,3]}), I(G_{[1,3]} > G_{[3,3]}), \\ I(G_{[3,3]} > G_{[3,1]}), I(G_{[3,1]} > G_{[1,1]})\}. \quad (10)$$

Fig. 2(e) shows the mapping results. Fig. 2(g) rewrite the mapping results in vector form, and we call it "marginal feature". In Fig. 2(h), central and marginal feature vectors are concatenated to form the final frame-level representation. From the whole procedure of extracting binary feature, the function sets $\mathcal{F}_c$ and $\mathcal{F}_m$ actually play the role of hash functions, which significantly reduce the computation (only eight times of integer comparisons) and storage cost (using one byte to represent one frame) of ordinal measure.

Same to the STF_CE, we use the w-$shingling$ concept to model the temporal structure of videos. The difference is only the primary central features are used to model the temporal structure, i.e. the w-$shingling$ of the central feature and w is set to 3. Consequently the visual shingle of the t-th key-frame is constructed by its own frame-level representation and two central features of the $(t+1)$-th and $(t+2)$-th key-frames. Figure 4 illustrates the procedure of representing a video from binary pattern to the LBP-based BoVS representation. Fig. 4(a) shows the binary representation of the video formed by sequentially concatenating all the frame-level representations, where t is the frame number, black solid rectangle represents value 0, and blank rectangle represents value 1. The first 4 bits represent the central feature and the second 4 bits represent the marginal feature. Fig. 4(b) depicts the visual shingle of the t-th key-frame, which is produced by combining the t-th key-frame (01000010) with the central feature of the $(t+1)$-th key-frame (0001) and the central feature of the $(t+2)$-th key-frame (1110). Fig. 4(c) shows the histogram of visual shingles, which is the BoVS representation for the original video.

Note that the design of spatiotemporal representation is very gen-

eral and we show particular implementation choices for our system through information theory and feature descriptor LBP. New spatiotemporal features can be obtained through changing the way of dividing frames, designing new ordinal relations selection algorithm, e.g., taking the labeling results of training dataset into consideration in the process of ordinal relations selection. In the experimental part, the efficiency and effectiveness of the STF_CE and STF_LBP based methods will be compared on two different sets.

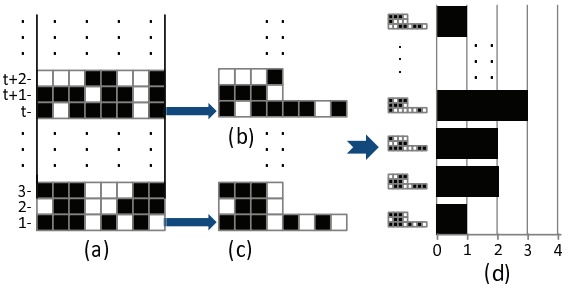

Figure 4: (a) An example video consisting of frames. (b) The visual shingle of the t-th key-frame. (c) The visual shingle of the first key-frame. (d) The BoVS representation.

4. INDEXING AND RETRIEVAL

In this section, we will present an efficient retrieval method by fast intersection kernel and inverted file. Let $\mathcal{S}$ denote the Web video dataset and its size is $|\mathcal{S}|$. Let M_i denote the number of extracted frames for the i-th dataset video, and the maximal value is denoted by $M^{\max}$. The value of $M^{\max}$ is 3248 for our constructed dataset. The number of extracted frames for query video is M_q. Let $\mathcal{V}$ denote the visual shingle vocabulary, whose size is denoted by $|\mathcal{V}|$. Query video V_q is represented by $[t_q^1, \cdots, t_q^w, \cdots, t_q^{|\mathcal{V}|}]$ and t_q^w is the number of times the w-th visual shingle occurs in the query video. Similarly, the i-th dataset video V_i is represented by $[t_i^1, \cdots, t_i^w, \cdots, t_i^{|\mathcal{V}|}]$, here $1 \le i \le |\mathcal{S}|$.

In this work, we adopt the histogram intersection

$$\mathrm{sim}(V_q, V_i) = \frac{\sum_w \min(t_q^w, t_i^w)}{\sum_w \max(t_q^w, t_i^w)} \tag{11}$$

to measure the similarity between query video and database videos, since this measurement does not involve computationally expensive multiplication as cosine similarity. To speed up near-duplicate video retrieval, we will present an exact and efficient histogram intersection calculation method based on fast intersection kernel [2] and inverted file [22].

The histogram intersection (11) can be rewritten as

$$\begin{aligned} \mathrm{sim}(V_q, V_i) &= \frac{\sum_w \min(t_q^w, t_i^w)}{\sum_w t_q^w + \sum_w t_i^w - \sum_w \min(t_q^w, t_i^w)} \\ &= \frac{\sum_w \min(t_q^w, t_i^w)}{M_q + M_i - \sum_w \min(t_q^w, t_i^w)}. \end{aligned} \tag{12}$$

where the first and second terms of denominator are fixed values. Therefore, the efficient calculation of (12) depends on how to reduce the number of comparison and summation operations involved in the numerator of (12). In the worst case, the times of comparison and summation operations are both $(|\mathcal{V}| \cdot |\mathcal{S}|)$. Fortunately, the BoVS representation is sparse, since the non-zero terms of BoVS representation is not larger than M_i and much less than the dimensionality $|\mathcal{V}|$, i.e. $M_i \le M^{max} \equiv 3248 \ll |\mathcal{V}| \equiv 65536$. Consequently the number of comparison and summation operations can both be reduced to $(M^{\max} \cdot |\mathcal{S}|)$ with the well-known indexing structure inverted file.

With the fast intersection kernel, we can further reduce the number of comparisons. The numerators of (12) was proven to be Mercer kernel [2] and called intersection kernel, whose fast calculation methods have been proposed for classification and clustering models. Maji et al. [18] presented an efficient intersection kernel Support Vector Machine(SVM). Wu et al. [25] applied fast intersection kernel to kernel k-means for visual codebook generation. The main idea of the fast intersection kernel is sorting the appearing times t_i^w in ascending (or descending) order and using a binary search to replace pairwise comparisons. So the number of comparisons is reduced from $(M^{\max} \cdot |\mathcal{S}|)$ to $(M^{\max} \cdot \log |\mathcal{S}|)$. Moreover, use the w-th visual shingle as an example, the set $\{t_i^w | 1 \le i \le |\mathcal{S}|\}$ contains at most $M^{\max}$ unique values, since $t_i^w \le M_i \le M^{\max}$. Thus, the time of binary search on the sorted list can further be reduced to $(\log M^{\max})$ with the help of the following modified inverted file.

Inverted file is a popular data structure used in retrieval systems [22]. Figure 5 shows the structure of our modified version. Same to the traditional inverted file which has an entry for each visual shingle followed by all the videos in which the visual shingle occurs. The difference is we arrange videos following the same visual term into different groups according to appearing times. Given a visual shingle S^w, $1 \le n_1^w < n_2^w < \cdots < n_{L_w}^w \le M^{\max}$ denote all the non-zero appearing times. Notations $V_{k,1}^w, \cdots V_{k,l}^w, \cdots, V_{k,I_k}^w$ denote the videos containing n_k^w number of visual shingle S^w. Pointer type variable p_k^w holds the address of the video V_{k,I_k}^w. The "w-th video ID array" (denoted as VID^w) records all the video IDs who have non-zero visual shingle S^w. Use the first tuple (n_1^w, p_1^w) as an example, since videos $V_{1,1}^w$ and $V_{1,2}^w$ contain the same number of visual shingle S^w, they are stored together and p_1^w holds the address of video $V_{1,2}^w$. For a query video V_q containing n_q^w visual shingle S_w, to calculate the similarity between query video and dataset video, a binary search is first conducted on sorted values $\{n_k^w | 1 \le k \le L_w\}$ to find the value r satisfying $n_r^w < n_q^w \le n_{r+1}^w$. Then we scan VID^w from the first entry and update similarities. The intersection kernel of V_q and $V_{k,l}^w$ is added by n_k^w for the videos from the first entry to the one pointed by p_r^w, and added by n_q^w for all the other videos of VID^w.

In the experimental part, we will conduct video retrieval experiments on the 50K Web video dataset to verify the modified inverted file can improve the retrieval efficiency in practice compared to the original inverted file.

5. EXPERIMENTS

In this section we evaluate the performance of our method on two Web video datasets: one is provided by CityU and CMU [5] and the other is constructed by us.

5.1 Datasets Description

In this paper, we are addressing the Web video retrieval problem, which has many different characteristics from the well-known TRECVID copy detection task [23]. First, the scalability challenge in TRECVID copy detection is weaker than Web video retrieval. For example, in TRECVID 2008, the dataset consists of 101 videos with a combined length of about 100 hours [16], and our larger dataset consists of about 50K and 3130 hours Web videos, which is about 30 times of the TRECVID dataset. Second, the two tasks have different distributions on the video transformation. For the

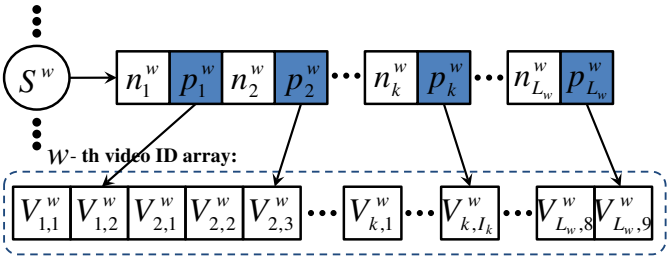

Figure 5: Modified inverted file.

TRECVID copy detection task, the duplicates are produced artificially by using a tool developed by IMEDIA [26], while in our experiments the duplicate transformations are all done by the real Web users, which reflects the real user behavior on generating duplicates. Since the objective of this paper is in particular to address the duplicate retrieval problem for Web videos, we adopted the public CC_WEB_VIDEO dataset and constructed a larger Web video dataset to evaluate our method.

CC_WEB_VIDEO Dataset was provided by **C**ityU and **C**MU. It consists of 24 sets of video clips (totally 12,790 video clips) downloaded from video sharing websites using specific keywords. For each set of videos, the most popular video is used as the query video, and two assessors were asked to manually label other videos within the set with the judgment "redundant" or "novel" to get the ground truth. For local feature based method, shot-based sampling method is used to extract key-frames and there are 398,015 key-frames are extracted in total. We adopted the uniform sampling to extract one frame per second and finally $2,636,471$ frames are extracted in total, which are about 6.6 times of that extracted in shot-based method.

Larger Web Video Dataset was constructed by adding videos crawled from Bing Videos to CC_WEB_VIDEO. There are 49,603 video clips in total. To our best knowledge, this is the largest published Web video dataset. The labeling results of CC_WEB_VIDEO is used as the ground truth of the new dataset. Key-frames were uniformly extracted per second and there are 11,270,825 key-frames in total, which is about 4.28 times of that used in the first experiment. We will provide the download links of this dataset once required by email.

5.2 Methods to be Compared

To evaluate retrieval quality, we present an extensive comparison of the proposed method and some existing content based near-duplicate video retrieval methods, which are briefly described as follows.

Color histogram based method: In [28], the authors used the color histogram method as a baseline method. The color histogram for each key frame is calculated as a 24 dimensional feature vector, which is concatenated with 18 bins for Hue, 3 bins for Saturation and 3 bins for Value. Each video is finally defined as a 24-dimensional vector of a normalized color histogram over all key-frames in the video.

Ordinal measure based method: A typical ordinal measure based method is [12], which developed a coarse-to-fine ordinal signature comparison scheme. In the coarse searching step, roughly matched positions were determined based on sequence shape similarity, and in the fine searching step, dynamic programming was applied to handle similarity matching in the case of losing frames.

Local feature based method: Wu et al. [28] proposed a hierarchical method to combining global signatures and local features. It first used color signature to detect near-duplicate videos with high confidence and filtered out very dissimilar videos. Then local feature based method was utilized to identify the remainder uncertainty videos.

LSH based method: In [6], Dong et al. extracted a HSV based color histogram to represent a key-frame and employed LSH to map color histogram to compact binary patterns.

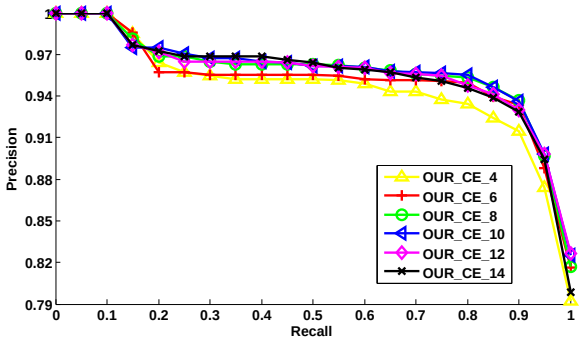

Figure 6: Averaged precision/recall graphs across 24 query videos.

5.3 Parameter Setting for STF_CE

To set the feature length K, we evaluate the performance of STF_LBP on the public CC_WEB_VIDEO dataset with different parameter settings ($K = 4, 6, \cdots, 14$). Notations OUR_CE_4, OUR_CE_6, $\cdots$, OUR_CE_14 are used to represent the CE-based method with K values $4, 6, \cdots, 14$, respectively. The precision and recall (PR) metric is adopted to evaluate effectiveness. Figure 6 depicts the PR curves. We can see that the CE-based methods have the similar PR curves for different feature numbers, so it is difficult to pick up the best setting for K through PR curves. The performance is further evaluated by mean average precision (MAP), time and storage costs. As in [6], we adopt the following definition of AP,

$$\text{AP} = \frac{1}{n}\sum_{i=1}^{n}\frac{i}{r_i} \tag{13}$$

where n is the number of relevant videos to query video, and r_i is the rank of the i-th retrieved relevant video. Experimental results are listed in the Table 2 and show OUR_CE_4 has the lowest MAP, OUR_CE_8 and OUR_CE_10 have the similar highest MAPs, OUR_CE_12 and OUR_CE_14 have lower MAPs than OUR _CE_8 and OUR_CE_10, although they extracted more informative features, which demonstrates preserving more ordinal relations is not sufficient to obtain better performance. In a real system, index should be able to be loaded in memory to enable fast search. If

we use four bytes to store video ID $V_{k,l}^w$, two bytes to store appearing time n_k^w and four bytes to store pointer type variable p_k^w. The method OUR_CE_14 has the highest time and storage costs. The method OUR_CE_8 has the highest MAP and relatively lower time and storage costs, so value 8 is a good setting for feature length K.

Table 2: Comparisons on time, storage costs, and MAP.

Methods	MAP	Time(ms)	Storage(MB)
OUR_CE_4	0.931	2.2	2.52
OUR_CE_6	0.941	2.8	3.92
OUR_CE_8	0.950	3.6	4.88
OUR_CE_10	0.950	4.0	6.08
OUR_CE_12	0.947	4.2	7.22
OUR_CE_14	0.946	5.5	8.16

5.4 Results on the CC_WEB_VIDEO Dataset

Let OUR_LBP represent the proposed method with STF_LBP. Let M_CH, M_OM, and M_HIER represent the color histogram based method, ordinal measure based method and the hierarchical method, respectively. To facilitate comparisons with existing methods, we execute each query against its corresponding video set as in [28]. We first evaluate the effectiveness of our methods (i.e. OUR_LBP and OUR_CE_8) and the three typical methods by PR metric. The PR curves of M_HIER and M_CH were provided by the authors of [28]. The implementation of M_OM is provided by the authors of [12] and all parameters are finely tuned. All the experimental results are shown in Figure 7. We can see that the proposed OUR_LBP and OUR_CE_8 have the similar PR curves and outperform all the other methods. Our methods have the similar precisions to M_HIER when recall is less than 0.70. For recalls larger than 0.70, our methods outperform M_HIER, since M_HIER only extracted one frame from each shot, which reduced the number of extracted local features while also broke the temporal continuity of videos. The ordinal measure based method is worse than our method for all recalls, since the proposed method implies a feature selection (using CE-based feature selection and LBP) on the ordinal features, and the method in [12] is sensitive to the settings of some parameters (e.g. weighting factors and distance thresholds). The performance of M_CH is poor, since M_CH only used the color properties without modeling spatiotemporal information and color histogram is highly susceptible to the global variations in color caused by different encoding devices [11].

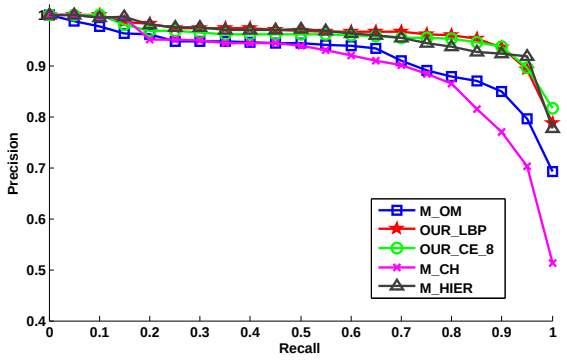

Figure 7: Averaged precision/recall graphs across 24 query videos.

The efficiency of the proposed methods is evaluated with average searching time over the 24 queries. Let M_LSH denote the LSH based method [6]. Table 3 shows the comparison results of our methods with M_CH, M_OM, M_HIER, and M_LSH from the aspect of MAP, time and storage cost. It can be seen that only M_LSH and the proposed methods (either OUR_LBP or OUR_CE_8) have real-time performance. The proposed methods have high efficiency, since we adopt the modified inverted file to calculate the histogram intersection between videos. The storage cost of the method OUR_LBP is 5.18MB. OUR_CE_8 has the similar storage cost, since the two methods have the same number of visual shingles and the only difference is the number of videos following each visual word. Thus, all index data can be loaded in memory, which confirms the efficiency. The high efficiency of the M_LSH method comes from the fast nearest neighbor search technique LSH. Same to the M_CH method, this method utilized the color histograms to represent a video, so it has the similar MAP to M_CH. The MAP and time cost of the method M_LSH were from [6]. Our methods and the local feature based method M_HIER achieve high MAPs (larger than 0.95) and significantly outperform M_LSH and M_CH. However, the time cost of M_HIER is about 2600 times of our method, since it still needs a large number of comparisons between key-frames. In Wu's recent work [29], they fused the contextual information (time duration, thumbnail image, etc.) with video content and achieved 164 times faster than their hierarchical model with a loss of performance. The proposed methods still perform at least 15 times faster than this state-of-the-art method. The speedup strategy used in [29] is to further reduce the number of key frames to be matched by incorporating contextual information, which can be directly transferred to other methods. While our methods are content-based, the contextual information can be incorporated into the proposed framework to further increase efficiency.

Table 3: Comparisons on time, storage costs, and MAP.

Methods	MAP	Time(s)	Storage(MB)
M_CH	0.892	-	-
M_OM	0.910	2.9	-
M_HIER	0.952	9.6	-
M_LSH	0.893	4.6×10^{-3}	-
OUR_LBP	0.953	3.7×10^{-3}	5.18
OUR_CE_8	0.950	3.6×10^{-3}	4.88

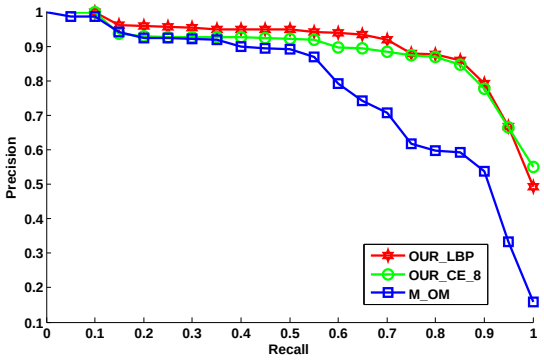

Figure 8: PR curves of our methods and M_OM.

5.5 Results on Larger Dataset

In this section we will evaluate the performance of the proposed method on the whole 50K Web video dataset. The proposed method will be compared to M_OM, since our binary frame-level representation is motivated by the ordinal measure. The searching result is

first evaluated by PR curves as depicted in Figure 8. We can see that for recalls less than 0.5, all the three methods achieve high precisions (larger than 0.9). At recalls larger than 0.5, our methods outperform M_OM and have higher precisions, which confirms our methods are effective for larger dataset. The proposed OUR_LBP slightly outperforms OUR_CE_8 for recalls smaller than 0.75, since LBP is a more general and robust feature descriptor. At the other recalls, the two methods have the similar precisions. Therefore, for larger dataset the heuristic method OUR_LBP slightly outperforms the automatic method OUR_CE_8 method, and STF_LBP has relatively better generalization ability than STF_CE.

The searching results of our methods are further evaluated by another metric normalized discounted cumulative gain (nDCG). nDCG is a retrieval measure devised specifically for Web search evaluation as it rewards relevant documents that are top-ranked more heavily than those ranked lower. Given a query video, the nDCG is computed as:

$$\text{nDCG@}N = Z_N \sum_{i=1}^{N} (2^{r(i)} - 1)/\log_2(1+i) \qquad (14)$$

where $r(i) \in \{0, 1\}$ is the relevance level of the result returned at position i and Z_N is a normalization constant that is chosen so that the optimal ranking's nDCG score is 1. Figure 9 shows the comparison of our methods and M_OM at points $N = 2^0, 2^1, \cdots, 2^{10}$. We can see that our methods outperform the M_OM method for different N values. All the eleven nDCG values of our methods are higher than 0.90, which demonstrates the high performance of the proposed methods from retrieval viewpoint.

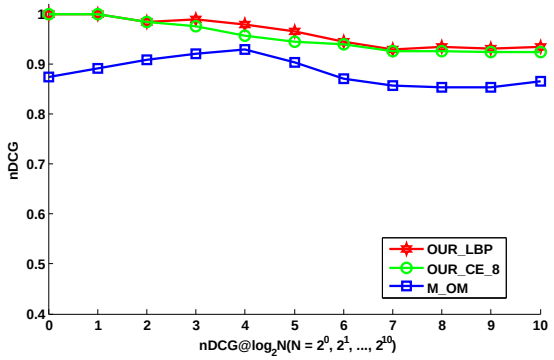

Figure 9: nDCG curves of our method and M_OM.

Table 4 shows the comparison results of our method with M_OM from the aspect of MAP, storage and time cost. For this larger dataset, our methods significantly outperform the M_OM method. The average time costs of our methods are smaller than 20.0ms and meet the real-time requirement. The time cost of M_OM is 243.4s, which is unacceptable high for search engine. With the same storage settings as the experiments on the CC_WEB_VIDEO dataset, the storage cost of OUR_LBP and OUR_CE_8 is 26.74MB and 25.63MB, respectively. Thus, the indices of our methods can still be loaded in memory for this larger dataset.

5.6 Memory Cost Analysis

For the used index structure, the memory cost consists of two main parts: appearing time tuples (n_k^w, p_k^w) and video ID arrays. Let SC^{vid} and SC^{att} denote the storage costs of video ID arrays and appearing time tuples, respectively. In this section, we will use the OUR_LBP method as an example to study the storage and time costs of our method for a 1M Web video dataset. To model the relationship between storage cost and dataset size, we randomly sample some video datasets from the 50K dataset. Let x denote the size of sampled dataset, $x = 5000, 10000, ..., 40000, 45000, 49603$. For each point x, we randomly sample five datasets with the same size x and use the average storage cost of the five datasets as the function value of this point. A log function $SC^{att}@x = 0.1228 \times \log(0.0036 \times x)$ is adopted to fit the relationship between SC^{att} and x, which is depicted in Figure 10(a). If the rule holds when 1M video dataset is used, the value of SC^{att} is still small, i.e. about 1.01MB. Figure 10(b) shows the relationship between SC^{vid} and x, which illustrates SC^{vid} is approximately proportional to x and the ratio is a small value 5.38e-4. We can calculate the SC^{vid} for 1M video dataset is 537.80MB. Therefore, theoretically a common PC with 1GB memory is capable of indexing 1M videos using the modified inverted file, which enables fast search on 1M video dataset.

Table 4: Comparisons on time, storage costs, and MAP.

Methods	MAP	Time(s)	Storage(MB)
M_OM	0.774	243.4	-
OUR_LBP	0.885	17.0×10^{-3}	27.38
OUR_CE_8	0.871	17.9×10^{-3}	25.63

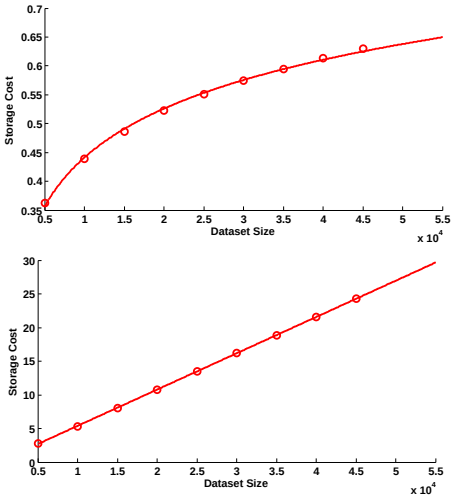

Figure 10: Relationship between storage costs and dataset size: The upper figure: $SC^{att}@x = 0.1228 \times \log(0.0036 \times x)$ The lower figure: $SC^{vid}@x = 0.0005378 \times x$.

5.7 Efficiency of the Modified Inverted File

The time complexity of calculating the histogram intersection between query video and database videos mainly consists of summation and comparison operations. Compared to the original inverted file, the modified inverted file reduces the comparison times from $(M^{\max} \cdot |\mathcal{S}|)$ to $(M^{\max} \cdot \log(M^{\max}))$ and it requires the same number of summations as the original one does. Although the theoretical time complexity (i.e. $M^{\max} \cdot |\mathcal{S}|$) of the modified inverted file is not improved, the actual comparison times is reduced. To verify the modified inverted file can improve the retrieval efficiency in practice, we implement the OUR_LBP method on the 50K dataset by replacing the modified inverted file with the original one. Experimental results show the modified inverted file (17.0ms) achieves 15.4% improvement on searching time compared to the original

one (20.1ms). This demonstrates that the modified inverted file can improve the retrieval efficiency by reducing the number of comparisons.

6. CONCLUSIONS AND FUTURE WORK

This paper proposed a spatiotemporal feature and constructed an efficient indexing structure to address the efficiency and scalability issues of near-duplicate Web video retrieval. Our frame-level binary signature leveraged relative gray-level intensity distribution within a frame. Temporal structure of video was modeled by the $w\text{-}shingling$ concept from text retrieval, which is an efficient method to modeling temporal information. The combination of fast histogram intersection kernel and inverted file provided an accurate and efficient calculation method for histogram intersection measurement between videos.

Compared to image retrieval, constructing a large-scale Web video dataset is more expensive, since crawling videos from Web needs more time and storage. Furthermore, the ground truth of near-duplicate videos is extremely expensive, since the assessors need to check the whole videos before giving their final judgement, and a glance is often sufficient to annotate two images. To our best knowledge, the 50K dataset is the largest published Web video dataset. However, it cannot be denied that this is still a small portion compared to the whole Web videos, so constructing an annotated large-scale Web video dataset is highly demanded. In future work, we will construct a 1M Web video dataset and evaluate our framework on it.

Acknowledgments

The authors would like to thank Chong-Wah Ngo and Xiao Wu for sharing the CC_WEB_VIDEO dataset with us.

7. REFERENCES

[1] D. N. Bhat and S. K. Nayar. Ordinal measures for image correspondence. *IEEE Trans. on PAMI*, 20(4):415–423, 1998.

[2] S. Boughorbel, J.-P. Tarel, and N. Boujemaa. Generalized histogram intersection kernel for image recognition. In *Proc. ICIP*, 2005.

[3] A. Broder. On the Resemblance and Containment of Documents. In *Proc. of the Compression and Complexity of Sequences*, 1997

[4] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: finding a (thick) needle in a haystack. In *Proc. CVPR*, 2009.

[5] CC_WEB_VIDEO: Near-Duplicate Web Video Dataset. Available: http://vireo.cs.cityu.edu.hk/webvideo/.

[6] W. Dong, Z.Wang, M. Charikar, and K. Li. Efficiently matching sets of features with random histograms. *ACM MM*, 179–188, 2008.

[7] M. Douze, A. Gaidon, H. Jegou, M. Marszatke, and C. Schmid. Inria-Learąś video copy detection system. In *TREVCID*, 2008.

[8] F. Fleuret and I. Guyon. Fast Binary Feature Selection with Conditional Mutual Information. In JMLR, 2004. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[9] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. VLDB*, 1999.

[10] K. Grauman and T. Darrell. The Pyramid Match Kernel: discriminative classification with sets of image features. In *Proc. ICCV*, 2005.

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

[12] X.S. Hua, X. Chen, H.J. Zhang. Robust video signature based on ordinal measure. In *Proc. ICIP*, 24–27, 2004.

[13] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008.

[14] J. Law-To, L. Chen, A. Joly, et al. Video copy detection: a comparative study. In *CIVR*, 371–378, 2007.

[15] J. Law-To, V. Gouet-Brunet, B. Olivier and B. Nozha. Local behaviours labelling for content based video copy detection. In *Proc. ICPR*, 232–235, 2006.

[16] J. Law-To, A. Joly, and N. Boujemaa. Muscle-vcd-2007: a live benchmark for video copy detection. *http://www.-rocq.inria.fr/media/civr/bench*, 2007.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machine is efficient. In *Proc. CVPR*, 2008.

[19] J. Matas, and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Proc. ICCV*, 2005.

[20] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *ACM MM*, 61–70, 2008.

[21] S. Siersdorfer, J. San Pedro, and M. Sanderson. Automatic video tagging using content redundancy. In *ACM SIGIR*, 19–23, July 2009.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 1470–1477, 2003.

[23] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proc. ACM MIR*, 2006.

[24] Ian H. Witten, and E. Frank. Data Mining: practical machine learning tools and techniques. Morgan Kaufmann, Amsterdam, 2005.

[25] J.X. Wu, and J. M. Rehg. Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *Proc. ICCV*, 2009.

[26] A. Joly, J. Law-to and N. Boujemaa. INRIA-IMEDIA TRECVID 2008: Video Copy Detection. In *NIST TRECVID Workshop*, 2008.

[27] P. Wu, T. Thaipanich, and C.-C. J. Kuo. A suffix array approach to video copy detection in video sharing social networks. In *Proc. ICASSP*, 3465–3468, 2009.

[28] X. Wu, C.-W. Ngo, and A. G. Hauptmann. Practical elimination of near-duplicates from Web video search. In *ACM MM*, 218–227, 2007.

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

[30] W.L. Zhao, S. Tan, and C.-W. Ngo. Large-scale near-duplicate Web video search: challenge and opportunity. In *Proc. ICME*, 2009.

[31] G. Zhao, and M. Pietikäinen. Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions. *IEEE Trans. on PAMI*, 29(6):915–928, June 2007.