[1]ON REAL-TIME DETECTING DUPLICATE WEB VIDEOS

Lu Liu[1], Wei Lai[2], Xian-Sheng Hua[2], Shi-Qiang Yang[1]
[1]DEPT of Computer Science and technology, Tsinghua University, [2]Microsoft Research Asia

ABSTRACT

With the rapid development of telecommunication techniques and digital devices, it is quite easy to copy, modify and republish videos in digital format, resulting in large volume of duplicate videos on the web in recent years. In this paper we mainly investigate the problem of detecting excessive content duplication, so as to facilitate video search and intelligence propriety protection. A real-time detection method is hence proposed, which first selects videos' representative frames and then reduces each to a 64 bit hash code. Then the similarity of any two videos can be estimated by the proportion of their similar hash codes. The experiments demonstrate that our approach is both efficient and effective in terms of real-time applications.

Index Terms— duplicate video detection, video signature

1. INTRODUCTION

With the rapid development of video capturing devices and online video hosting services, the amount of videos grows explosively on the web in recent years. For example, the popular video sharing site YouTube [1] announced that about 100 million videos were served per day while Yahoo [2] indexed more than 50 million audio and video files. Due to the prevalence of video edit tools, it is quite easy to copy, modify and republish video files in digital format, resulting in duplicate contents spreading over the Internet. As evidenced, each video in Cheung's collection has around five similar copies on average [3]. Table 1, which summarizes the duplicate number in the first page after we input some well-known queries to Yahoo video search engine, also illustrates this issue. Duplicate video identification will benefit video search, save storage, facilitate intelligence propriety protection, provide an alternative copy in case of expired links and present the best version based on users' need as well [3].

Table 1 Results from Yahoo video search engine

Query	Headbutt	Bill Gates	Connie Chung
Duplicate number	2	3	3

How to select the most representative and dispersive frames of the videos and then reduce their high dimensional features to a compact representation are the key problems in the fast video duplicate detection on a large-scale database. A rich literature of previous works typically employed the temporal trace, such as the sequence of key frames of shots or the high-rate sampled frames, to represent a video [4-8]. This kind of representations is highly sensitive to temporal changes and correlated with the duration of videos, which brings difficulties for signature comparison. To avoid the problems, k-medoid method in [9] and seed vector method in [3][10] are proposed to select a predefined number of representative frames. For example, seed vector method in [3][10] generates a fixed number of seed vectors firstly and then selects the frames closest to each seed vector as the signature frames for further comparison. This scheme is robust to temporal changes and uncorrelated with video duration. However, the high-dimensional features of representative frames make the comparison too slow to be applicable for real-time applications.

In this paper, we propose an efficient scheme to detect duplicate videos on the Web. First, each video's representative frames are selected. Aiming at make our scheme uncorrelated with duration and robust to temporal changes, we utilize the seed vector method in [3][10] to select representative frames. We also propose three metrics to evaluate the effectiveness of seed vectors and select optimal set of seed vectors accordingly, which significantly improves the performance of duplicate detection. Then these frames are converted to K-bit hash codes by PCA and vector quantization for reference to [11]. At last, the similarity of any two videos can be estimated by the proportion of their similar hash codes. In the scheme, we use the first L bits of hash codes to index all the videos so that the comparison time is reduced greatly. Furthermore, as the core comparison is finally carried out in a bit "exclusive or" operation, the video comparison can be fast enough to meet the real-time need. The experiments demonstrate that only about 0.98ms is required for one video's duplicate search in the database with 10000 videos.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the concept of duplicate videos in the scope of this paper, followed by the videos' hash codes generation scheme in Section 3. Section 4 presents the real-time detection method. Section 5 proposes three metrics to evaluate seed vectors to improve the seed vector generation method. Experimental results will be discussed in Section 6 and we conclude this paper in Section 7.

2. DUPLICATE VIDEO DEFINITION

Based on the survey and analysis of real web videos, we clarify the concept of duplicate videos in the scope of this paper. Duplicate videos on the web are with roughly the same content, but may have three prevalent differences.

[1] This work was performed while Lu Liu was visiting Microsoft Research Asia as a research intern.

- **Format:** there are many video formats on the web nowadays such as MPG, AVI and WMV, etc.
- **Bit-rates, frame-rates, frame size:** in order to facilitate storage, downloading, streaming or to meet other needs of users, the same video may be compressed at different qualities, reformatted to different frame-rates and sizes.
- **Editing in either spatial or temporal domain:** spatial and temporal editing are ubiquitous on web videos, e.g. logo appears on the top or bottom corner for different sources, a few frames are dropped due to network congestions, or a short clip is inserted into the original stream.

3. HASH CODES GENERATION

In this section, we describe the whole process of video hash codes generation as illustrated in Figure 1.

3.1 Video Re-sampling

The content within a second is most likely consistent in most videos. Therefore, we re-sample the videos uniformly at 1 frame per second (fps) and represent them as the sequence of sampled frames as below,

$$V = \{f_k, 1 \le k \le FN(V)\}. \quad (1)$$

where *FN(V)* is the number of sampled frames.

3.2 Representative Frames Selection

Given a set of seed vectors $S_{SV} = \{sv_i, 1 \le i \le m\}$, the representative frames set *SF* of the video *V* is defined as

$$SF \triangleq (sf_1, sf_2, ..., sf_m)$$
$$where \quad sf_k = \arg\min_{f_j \in V} dis(f_j, sv_k) \quad sv_k \in S_{SV} \quad (2)$$

where *dis* is the Euclid distance. f_j, sv_k, sf_k are *N*-dimensional feature vectors. Intuitively *SF* consists of the frames in *V* which are closest to the corresponding seed vectors in S_{SV} [3].

3.3 Dimension Reduction and Hash Codes Generation

As long as the representative frames are selected, their feature dimension is reduced by principle component analysis (PCA) to the most important *K* dimensions. Suppose the *i*-th representative frame in *SF* is represented as

$$sf_i' = \{sf_i^1 \quad sf_i^2 \quad ... \quad sf_i^K\} \quad (3)$$

where sf_i^j is *j*-th dimensional value of sf_i sorted descending by the eigenvalues. In order to further reduce the comparison computation complexity and storage cost, we transform sf_i' into a *K*-bit binary string SH_i as below:

$$SH_i^{\,j} = \begin{cases} 1 & if \quad sf_i^j > mean_j \\ 0 & if \quad sf_i^j \le mean_j \end{cases} \quad 1 \le j \le K \quad (4)$$

where $SH_i^{\,j}$ is *j*-th dimensional value of SH_i, $mean_j$ is the mean value of dimension *j*. In this way, each video is reduced to *m*K* bits (*m* is the number of seed vectors) represented as

$$VS = \{SH_1, SH_2, ..., SH_m\} \quad (5)$$

The compact representation requires very SMALL storage. For example, suppose $m = 50$, $K = 64$ as in our experiments, 1 million videos only need $1 \times 10^6 \times 50 \times 8$byte = 400M.

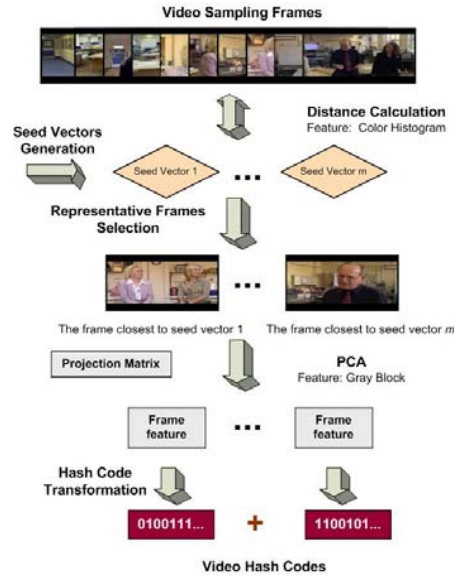

Figure. 1. Video Hash Codes Generation

4. DUPLICATE DETECTION

After each video's hash codes are generated, their similarity can be estimated by the proportion of their similar hash codes. In this Section, we explain how to search duplicate videos in our scheme.

Suppose two videos are reduced as $RV = \{RH_1, RH_2, \ldots, RH_m\}$, $QV = \{QH_1, QH_2, .., QH_m\}$. The difference between their *i*-th hash codes can be indicated by Hamming distance as below:

$$Hamming(RH_i, QH_i) = \sum_{k=1}^{K} (RH_i^{\,k} \oplus QH_i^{\,k}) \quad (6)$$

where $\oplus$ is the "exclusive or" operator.

The corresponding representative frames in the duplicate videos are similar with [3] but not the same. Therefore, their hash codes would not be identical but with a little difference. Due to principle of PCA, the difference is more likely to occur in less significant dimensions than in more significant ones. Therefore, we define the concept that two hash codes are deemed to be similar as below:

$beSim(RH_i, QH_i)$ is true if

$$\sum_{k=1}^{L} (RH_i^{\,k} \oplus QH_j^{\,k}) = 0 \ \ and \ \ \sum_{k=L+1}^{K} (RH_i^{\,k} \oplus QH_j^{\,k}) \le T_h \quad (7)$$

That means if the hash codes are identical at the most important *K* bits and have the distance less than T_h at the less important *K-L* bits, they are deemed as similar.

Then we can define the similarity of two videos as

$$Sim(RV, QV) = \frac{\sum_{i=1}^{m} I(beSim(RH_i, QH_i))}{m} \quad (8)$$

$I(A)$ equals to 1 if A is true and 0 otherwise. Suppose T_v is the video hash codes similarity threshold, that means the videos RV,QV are deemed as duplicate if and only if $Sim(RV,QV) > T_v$. Both T_h and T_v are tunable thresholds for different application scenarios.

Based on the formal video duplicate definition above, we build m hash tables, the i-th of which maps the videos whose i-th hash codes have the same first L bits to the same set, so as to reduce the search range GREATLY. For example, suppose $L = 16$, then each hash code only needs to compare with the corresponding one of other $1 \times 10^6/2^{16} =$ 15 videos in the database with 1 million videos. If we utilize multi-core possessors, the m hash codes could be compared at the same time. At last, the videos with more than $T_v * m$ similar hash codes are deemed as duplicate.

5. SEED VECTOR GENERATION

In our scheme, we select the frames closest to a set of seed vectors to be videos' representative frames [3] [10]. Cheung used a four-step algorithm to generate random seed vectors [10]. We improve the algorithm and employ a heuristic method resembling to [12] to select seed vectors incrementally, in which, a criterion is needed to judge seed vectors' effectiveness. Therefore, we propose three metrics to measure three aspects of the seed vectors as follows.

Suppose the S_{TV} is the training video set. In order to reduce the repetitive search space, all the feature vectors of frames in the S_{TV} are first clustered by K-means, and the cluster centers compose the training feature set $S_{TF} = \{tf_i, 1 \le i \le l\}$. Suppose SS_{SV} is the set with seed vectors already selected.

5.1 Validity Metric

First, we define the distance between video V and seed vector sv_j as below

$$dis(V, sv_j) = \min_{f_k \in V} dis(f_k, sv_j) \tag{9}$$

Where dis is the Euclid distance metric. Intuitively the seed vector should be close to videos based on the distance defined above. Take Figure 2 for example. Two duplicate videos distinguished by different colors have two frames represented by the circle. If the seed vector is far from the videos, two total different frames may be selected as the representative frames as shown in the figure. Therefore, in order to avoid the noise, we define the validity metric as

$$VM(tf_k) = 1 - \frac{1}{|S_{TV}|} \sum_{V \in S_{TV}} \min_{sv_j = tf_k \| sv_j \in SS_{SV}} dis(V, sv_j) \tag{10}$$

5.2 Diversity Metric

As close seed vectors have similar effectiveness so that it wastes the storage and slower the comparison, the seed vectors should be diversified in the feature space. So we should select the seed vector that is almost orthogonal to current selected seed vector set. The diversity metric is defined as

$$DM(tf_k) = 1 - \max_{sv_i \in SS_{SV}} \cos(tf_k, sv_i) \tag{11}$$

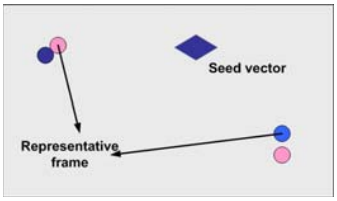

Figure. 2. Seed vector far from videos brings on noise

5.3 Salience Metric

Different seed vectors have different effectiveness or contributions. The feature vectors in the cluster which contains more videos should be more "important". Therefore, we define the salience metric as follows:

$$SM(tf_i) = \frac{VN(tf_i)}{|S_{TV}|} \tag{12}$$

where $VN(tf_i)$ is the number of videos in the cluster which contains tf_i.

We use the three metrics' product as the combining metric, based on which a heuristic method [12] is employed to select seed vectors incrementally.

6. EXPERIMENT RESULTS

In this section, we present the experiments results on the web videos to demonstrate the performance of our approach.

We randomly choose about 600 web videos as T_{TV}, from which the seed vectors are selected. In the step of representative frames selection, the 2 by 2 block color histogram (256 dimension), while in the step of hash codes generation, the combination of 8 by 8 and 7 by 7 gray block (113 dimension) is used.

As the numbers of web videos' duplicate copies are not easy to obtain, we first simulate web circumstances to produce videos' duplicate copies as our test data with ground-truth to measure both precision and recall of our approach. Then, we do the experiments on real web videos and check the results by human to get only precision rate.

In the first experiment, we simulate the web video duplicate cases and randomly produce zero to five duplicate copies of another 600 web videos, which don't overlap with the training data. These changes include the random combination of difference described in Section 2. Let A denote the number of video pairs found to be duplicate by using our approach, among which, B pairs are truly duplicate. Then precision is defined as B/A. Recall is defined as B/C where C is the total number of truly duplicate pairs in the ground-truth set. L is set 16.

First, we change the value of T_H with T_V equals to 0.15. Then we increase T_V with T_H equals to 30. The result curves in Figure 3 and Figure 4 show that precision is robust to

both T_H and T_V, but recall decreases when the criterion becomes stricter.

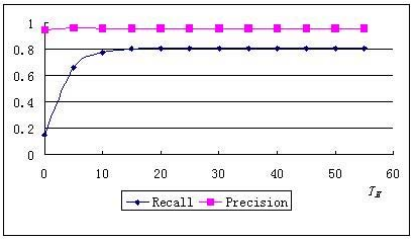

Figure.3 Precision and Recall changes with T_H

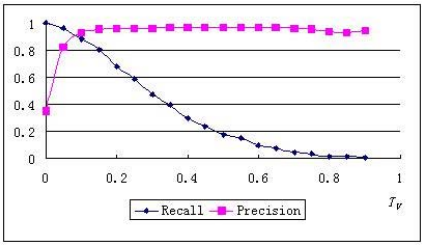

Figure 4 Precision and Recall changes with T_V

Then we test how the number of seed vectors influences the performance. The results in Figure 5 show that both precision and recall are improved along with the number of seed vectors increasing.

In order to measure the effectiveness of our metrics, we compare the result of our seed vector generation method with the one of original method in [10].The results shown in Table 2 demonstrate that our method can improve both precision and recall.

In the second experiments, we test our approach in more than 10000 real web videos. In this experiment, we use each video in the database as query sample to get its possible duplicate copies based on the parameter $K = 64$, $L = 16$, $T_H = 15$ and $T_V = 0.2$. After deleting overlap cluster, we get 112 possible duplicate clusters at last, among which 90 clusters are determined to be real duplicate by human's checking. Therefore, precision is 80.36%. At the same time, we record the total search time, and get that each video's duplicate search needs 0.928ms. Therefore it demonstrates that our method can meet the real-time need.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose a real-time approach to detect web videos duplicates. Each video's representative frames are selected and reduced to 64 bit hash codes by PCA and vector quantization. The similarity of videos is estimated by the proportion of their similar hash codes. The experiments results demonstrate that our approach is both efficient and effective in terms of real-time duplicate video detection.

In future work, better approach to select representative frames will be employed to make our approach more robust for web video search engines.

Table 2 Result Comparison

$T_H = 30$	$T_V = 0.15$		$T_V = 0.10$	
	precision	recall	precision	recall
Original Method	0.9297	0.7722	0.9137	0.8518
Our Method	0.9519	0.8072	0.9321	0.8822

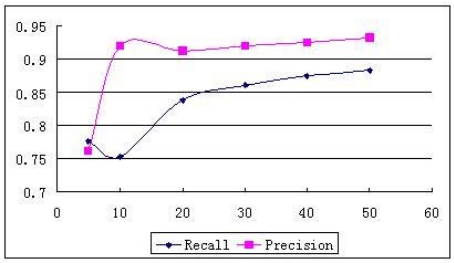

Figure 5 Precision and Recall changes with seed vector number

8. ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China under Grant No.60573167 and Tsinghua-Microsoft Cooperation Project.

9. REFERENCES

[1]http://mashable.com/2006/07/17/youtube-hits-1-million-videos-per-day/

[2]http://www.searchenginejournal.com/index.php?p=2036

[3] SC.Cheung, , et al.: "Estimation of Web Video Multiplicity". Internet Imaging, pages 34-36, San Jose, California. January 2000.

[4]H. Aoki, et al, "A Shot Classification Method of Selecting Effective Key-frames for Video Browsing", ACM Multimedia, pp. 1-10, 1996.

[5]N, Dimitrova, et al, "Content-Based Video Retrieval by Example Video Clip", IS&T and SPIE Storage and Retrieval of Image and Video Data-bases, vol. 3022, pp. 184-196, 1998.

[6]X. S. Hua, et al, "Robust Video Signature Based on Ordi-nal Measure", International Conference on Image Processing, Singapore, October 24-27, 2004.

[7]Z. Li, et al, "Fast video shot retrieval based on trace geometry matching", Vision, Image and Signal Processing, IEE Proceedings-vol. 152, issue 3, pp. 367-373, Jun. 2005.

[8]Y. P. Tan, et al, "A Framework for Measuring Video Similarity and its Application to Video Query by Example", IEEE Int. Conf. on Image Processing, 1999.

[9]H. Chang, S. Sull, and S. Lee, "Efficient video indexing scheme for content-based retrieval.", IEEE Trans. Circuits Syst. Video Technol, vol. 9, pp. 1269–1279, Dec. 1999.

[10]SC. Cheung, A. Zakhor, "Efficient Video Similarity Measurement with Video Signature", IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, issue. 1, Jan, 2003.

[11]B. Wang, et al, "Large-Scale Duplicate Detection For Web Image Search", Proceedings of IEEE Internation Conference on Multimedia & Expo, pp. 353-356, Toronto, 2006.

[12]Y. Wu, et al, "Sampling Strategies for Active Learning in Personal Photo Retrieval", Proceedings of IEEE International Conference on Multimedia & Expo, pp. 300-311, Toronto, 2006.