Detection and Location of Near-Duplicate Video Sub-Clips by Finding Dense Subgraphs

Tianlong Chen[1], Shuqiang Jiang[1], Lingyang Chu[1], Qingming Huang[1,2]

[1] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China

[2] Graduate University of Chinese Academy of Sciences, Beijing, 100049, China

{tlchen, lychu, qmhuang}@jdl.ac.cn , sqjiang@ict.ac.cn

ABSTRACT

Robust and fast near-duplicate video detection is an important task with many potential applications. Most existing systems focus on the comparison between full copy videos or partial near-duplicate videos. While it is more challenging to find similar content for videos containing multiple near-duplicate segments at random locations with various connections. In this paper, we propose a new graph based method to detect complex near-duplicate video sub-clips. First, we develop a new succinct video descriptor for keyframe match. Then a graph is established to exploit temporal consistency of matched keyframes. The nodes of the graph are the matched frame pairs; the edge weights are computed from the temporal alignment and frame pair similarities. In this way, the validly matched keyframes would form a dense subgraph whose nodes are strongly connected. This graph model also preserves the complex connections of sub-clips. Thus detecting complex near-duplicate sub-clips is transformed to the problem of finding all the dense subgraphs. We employ the optimization method of graph shift to solve this problem due to its robust performance. The experiments are conducted on the dataset with various transformations and complex temporal relations. The results demonstrate the effectiveness and efficiency of the proposed method.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval model

General Terms

Algorithms, Experimentation.

Keywords

Near-duplicate video detection, Sub-clip location, Graph shift

1. INTRODUCTION

With the rapid development of internet and multimedia technologies, the amounts of videos are growing explosively. Among these giant amounts of videos, there exist many duplicate video clips [16]. Detecting such near-duplicate video clips is desired by many potential applications such as web video retrieval [5, 9], automatic video tagging [5, 14] and large scale video database mining [13]. The common steps of this task can be summarized as follows. First, keyframes are extracted from videos by uniform sampling [7, 9] and shot-based method [5, 12]. Second, the keyframes are represented by a set of high dimensional feature vectors, including global features [7, 15] and local features [10, 11]. Finally, the similarity between two videos is determined by comparing the key frames in a pair wise manner.

Most existing systems focus on matching two full copy videos [9, 12, 13, 14] or partial near-duplicate videos [5, 6, 7, 15]. However, two similar videos may contain multiple near-duplicate segments in random locations due to temporal editing, such as inserting in or cutting out. Lifeng Shang *et al*. [9] constructed an efficient indexing structure and proposed a spatiotemporal feature to address the issues of real-time large scale near-duplicate web video retrieval. Their approach focuses on the running speed and accuracy, but fails to locate the similar sub-clips. The method of [15] focused on the robustness and uniqueness of the video signature, it propose a coarse-to-fine signature comparison scheme to locate a short query video clip in a long target video. Tan *et al*. [5] [6] presented an approach to detect and locate near-duplicate segments from two videos through the joint consideration of visual features and temporal coherency of frame sequence. All these methods perform not so well in the challenging situation when there are complex connections of similar segments from two videos. As shown in Figure 1, the near-duplicate segment connections could be one-to-one, one-to-many or many-to-many, and the order of those near duplicate sub-clips could be completely random.

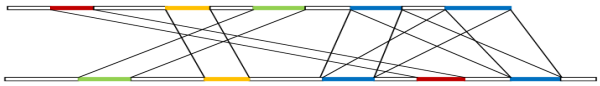

Figure 1. Near-duplicate videos containing complex connections.

In this paper, we propose a graph based method to robustly handle the challenging situation of videos with complex similar sub-clip connections. The main procedures are illustrated in Figure 2. After keyframe sampling and frame matching, a graph is established whose nodes represent matched keyframe pairs. By setting the temporal constraints into the edge weights, detecting complex near-duplicate sub-clips is transformed to the problem of finding all the dense subgraphs. We solve this optimization problem by using the method of graph shift [2]. Due to the intrinsic property of graph shift, our method is capable of detecting various conditions of near-duplicate sub-clip connections such as one-to-one, one-to-many or many-to-many. The proposed approach is fast and simple to implement. Compared with existing methods of partial-duplicate video detection, our approach has the following contributions:

1.The problem of near-duplicate sub-clip detection is converted to finding dense subgraphs.

*Area Chair: Lexing Xie

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

Copyright 2011 ACM 978-1-4503-0616-4/11/11…$10.00.

2.We use a new succinct frame representation method called weighted histogram of gradient for similar key frame matching.

3.The boundaries of the complex similar sub-clips at random positions could be effectively located via subgraph analysis.

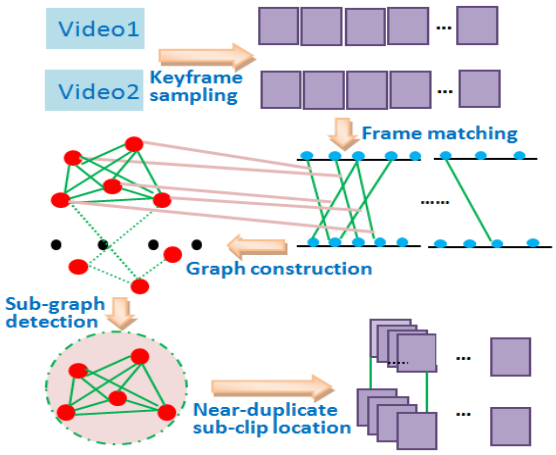

Figure 2. Illustration of the flowchart of our approach

2. WEIGHTED HISTOGRAM OF ORIENTED GRADIENT

In this paper we use a novel frame-level descriptor for video, combining the Histogram of Oriented Gradient (HOG) [8] and the Relative Mean Intensity (RMI) together by means of a weighting scheme. A frame is partitioned into rings which are invariant for the transformations such as rotation and flipping. Besides, instead of treating each frame as a whole, using a series of rings can save the local patterns and further make the descriptor more discriminative. RMI of each ring represents the bottom physical feature of a frame, and HOG, which is well-known for counting occurrences of orientation in localized portions of an image, has been improved in this paper with RMI as the weight. Compared with existing representations of video, the proposed descriptor offers several advantages. First, the combination of two naive features is succinct in concept. Second, it is compact in structure for encoding each frame at a certain sample rate. Third, it is invariant for common transformations such as: lighting, flipping, rotation, etc. Finally, it is also fast in extraction and matching procedure.

3. GRAPH CONSTRUCTION AND SUB-CLIP DETECTION

The idea of the graph model has been applied to various applications, such as image retrieval, image annotation and data clustering [2]. Liu at el. successfully used robust Graph model to detect common visual patterns in images [3]. In this paper, we extend the idea of finding dense subgraph in the temporal domain to detect and locate near-duplicate video sub-clips. A dense subgraph is a strongly connected subset of vertexes in a weighted undirected complete graph. The undirected graph is constructed by firstly using the detected similar keyframe pairs as the graph node, then associating the evaluation of temporal consistence and frame similarity with the edge weight. In this way, the validly matched frame pairs that are temporally consistent would form a dense subgraph and could be robustly detected by the method of graph-shift [3]. Thus the task of near-duplicate video sub-clip detection could be solved by detecting all the dense subgraphs in the previously mentioned complete graph.

3.1 Graph Construction

Given two sequences of keyframe points extracted from two videos, $V=\{v_1,v_2,...,v_n\}$ and $Q=\{q_1,q_2,...,q_m\}$, where m and n are the key frame numbers of V and Q respectively. Each keyframe is represented as a weighted histogram of oriented gradient as introduced in the last subsection. We conduct keyframe-pair comparison in the product space $P=V\times Q$, and obtain $n\times m$ keyframe pair, for a keyframe pair $p_i=(v_{i_1},q_{i_2})$ $v_{i_1}\in V$ $q_{i_2}\in Q$, we can calculate the feature similarity value by histogram intersection [9]. The function of the similarity value can be denoted as $S_{p_i}=f_1(i_1,i_2)$. The similarity value below a threshold ε means the two keyframes are not matched. The threshold parameter can be set relatively relax to include as much matched pairs as possible. In the experimental section, we will conduct an evaluation analysis on different threshold settings.

Given two keyframe pairs from two video, $p_i=(v_{i_1},q_{i_2})$ and $p_j=(v_{j_1},q_{j_2})$, we can compute the distance d_1 between keyframes v_{i_1} and v_{j_1} in the first video keyframe sequence. Likely, we can obtain d_2 between q_{i_2} and q_{j_2} in the second sequence. As there may exist various transformations in videos, and the frame per second may also change, we use the parameter s to represent the ratio between the two videos' frame rate. Now we can define a nonnegative monotonic decreasing function to encode this temporal information of keyframes.

$$T_{p_ip_j}(s)=f_2(|d_1-sd_2|) \tag{1}$$

Given two keyframe pairs, in other words, two nodes of the graph, the weight of the nodes is defined as

$$w_{ij}(s)=S_{p_i}S_{p_j}T_{p_ip_j}(s) \tag{2}$$

Suppose the set M contains k frame-pairs, and the graph G is $k\times k$ matrix. The adjacency matrix representation $A(s)$ for the graph G is as follows:

$$A(i,j)(s)=\begin{cases}0, i=j\\ w_{ij}(s), i\neq j\end{cases} \tag{3}$$

Obviously the graph G is symmetric and nonnegative. Suppose a near-duplicate segment has r keyframe-pairs. It corresponds to a dense subgraph T of G with r vertices, which is a weighted countpart of maximal clique. If we represent a subgraph by a vector x, $x\in\Delta$, where $\Delta=\{x\in R^m : x\geq 0 and\ |x|_1=1\}$. x_i, the i-th component of x, denotes the probability of this subgraph contains the vertex i. the modes of a graph G are approximately local maximizers of graph density $g(x)=x^TA(s_0)x$. Given a vector x, the corresponding subgraph is $G(x)$. If x^* is a local maximizers of g(x), the $G(x^*)$ is a dense subgraph which is also a near-duplicate sub-clips. Therefore we need to calculate all local maximizers of this quadratic function:

$$\text{Maximize}\quad g(x)=x^TA(s_0)x\quad \text{Subject to}\quad x\in\Delta \tag{4}$$

The solution x^* is sparse. In the same solution, we can find that some components have relatively larger values, while other smaller values may correspond to noises or outliers. Therefore we can detect clusters according to the value of x^*, so the clusters may correspond to the near-duplicate segments.

3.2 Detection and Location of Near-duplicate Sub-clips

Given an initialization *x(0)*, the corresponding local solution x^* can be efficiently calculated. In [2], the authors proposed an algorithm to find all large local maximizers $\{x^*\}$. For a local maximizer x^*, they also presented an algorithm to find key pattern L from x^* in [3]. The algorithm is based on the fact that x^* is sparse and the number of its sufficiently large components usually indicates how large the key pattern is. For a key pattern L which consists of many matched frame-pair, we can detect the clusters according to the frame sequence number. Then we can discover the boundaries of the clusters to locate the boundaries of the similar segments.

The proposed method has the following advantages. First, temporal pattern detection is converted to dense subgraph seeking. The dense subgraph is a strongly connected subset of vertexes in a graph. According to the definition of the weight of graph edge in section 3.1, the correspondence between two frame pairs from the same similar connection is stronger than that from different connections or noises. Therefore this conversion is reasonable. Second, our approach could handle the complex situation shown in Figure 1. A dense subgraph only corresponds to a similar segment-pair. When two similar videos contain complex connections, there exists the same number of dense subgraphs corresponding to these similar sub-clips. The problem of finding dense subgraphs could easily be solved by the graph shift algorithm. Finally, the dense subgraph based method can accurately locate sub-clips. The localization of similar segments boundaries is through the analysis on the vertex set from the dense subgraph. This vertex set has less or no outlier, so we can obtain high overlap of the similar sub-clips from two videos.

The time complexity of keyframe matching is $O(n \times m)$ and $O(k \times k)$ for computing $A(s)$. A matching window is used to accelerate the matching of two video clips by aligning the two keyframe sequences [4]. However, it is less effective for the near-duplicate segments with random positions. The method of finding dense subgraphs is computationally efficient as discussed in [3]

4. EXPERIMENTS

To demonstrate the effectiveness and efficiency of our method, we conduct experiments in the dataset of 20 groups of videos. Every group consists of one query video and four reference videos which are all about ten minutes in length. The videos are edited manually consisting 4~6 sub-clips which is from TRECVID [1] with various transformations. These similar sub-clips may be sequential or cross at random temporal positions. The lengths of these sub-clips vary from tens of seconds to 3 minutes. We use two measures to evaluate the performance of our method: average precision and average recall. The precision and recall of each group is denoted as follows [6]:

$$\mathrm{Recall} = \frac{\#Correctly\ \ detected\ \ near-duplicate\ \ frames}{\#Total\ \ near-duplicate\ \ frames}$$

$$\mathrm{Precision} = \frac{\#Correctly\ \ detected\ \ near-duplicate\ \ frames}{\#Detected\ \ near-duplicate\ \ frames}$$

The nonnegative monotonic decreasing function f_2 in section 3.1 is denoted as:

$$f_2 = \exp(-\frac{|d_1 - sd_2|^2}{\phi})$$

In our experiments, ϕ =100.

4.1 Near-duplicate Video Sub-clips Detection

A total of 80847 keyframes are extracted from the whole database. We evaluate the sensitivity of the system with different parameter setting. The experiments are conducted by varying the respective parameters while fixing the others to ε =0.95, η =0.01, φ =0.83. η and φ have been defined in [3], where η is the threshold to merge two similar local maximizers and φ is the threshold to drop some local maximizers with small $g(x^*)$. From Figure 3, we can see that, ε has a significant impact on the results. The average precision reaches the highest in ε =0.95 while the average precision is highest in ε =0.94. In general the precision and recall are not sensitive to η and φ. When the value of φ is higher than 0.83 or η is too small, the performance will be affected.

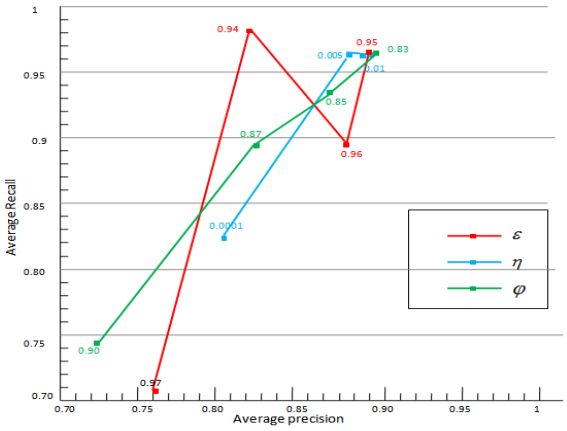

Figure 3. Sensitivity to parameters

From the above parameter analysis, we select these parameters to compare the performance of our method to a modification of Sub-Maximum Size Matching (SMSM) [7]. SMSM constructs a bipartite graph to identify the most suitable 1:1 mapping by optimizing the factors of visual content, temporal order and frame alignment together. We first extract dense segments of the first video and then apply SMSM. Table 1 shows the performance comparison. Our method outperforms SMSM in both average precision and average recall. This is because SMSM weakly handles videos containing complex connections.

Table 1. Result of method comparison

Approaches	Average Precision	Average Recall
Our method	89.00%	96.427%
SMSM	87.68%	85.40%

4.2 Complex Example

In this section, we use two long videos containing complex near duplicate sub-clip connections to demonstrate the performance of our approach. There are 2073 keyframes extracted from the first video, and 2394 key frames from the second video. The two videos containing complex sub-clips are shown in Figure 4(a). The numbers in the Figure represent the frame number. Then we use our approach and SMSM to detect the near-duplicate segments. The parameters in our approach are set as: ε =0.95, η =0.01, φ =0.83. The time of frame-pair comparison is 51s and the time of seeking temporal pattern is 57s. From Figure 4(b), lines with the same color denote the similar segments. We can see that our approach detects seven similar segments totally, six of which are correct, and the overlap is nearly 100% of the correct segments. The incorrect one is connected by two dotted lines. Figure 4(c) shows the result of SMSM. We can see our approach has higher overlap and precision. The experiments on many other video data also get similar results.

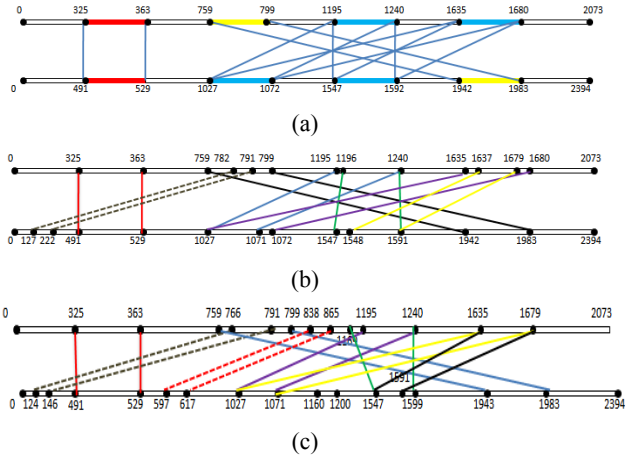

Figure 4. Illustration of complex example. (a) videos containing complex connections. (b) the result of our approach (c) the result of SMSM

5. CONCLUSION

In this paper, we present an effective solution to detecting and locating complex near-duplicate video sub-clips by finding dense subgraphs. The visual and temporal information are integrated and encoded to the established graph. The visual information is obtained by matching keyframes which are represented as the weighted histogram of oriented gradient, and the temporal information is taken into account in the weight of graph edge. In future work we shall try to speed up the computing time of the frame match.

6. ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China: 61025011, 60833006 and 61070108, in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by Beijing Natural Science Foundation: 4092042.

7. REFERENCES

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

[2] H.-R. Liu and S.-C. Yan. Robust Graph Mode Seeking by Graph Shift. *In ICML, 2010.*

[3] H.-R. Liu and S.-C. Yan. Common visual Pattern Discovery via Spatially Coherent Correspondences. *In CVPR, 2010.*

[4] H.-K. Tan, X. Wu, C.-W. Ngo and W.-L. Zhao. Accelerating near-duplicate video matching by combining visual similarity and alignment distortion. *In ACM Multimedia, 2008.*

[5] H.-K .Tan, C.-W. Ngo, Richang Hong and T.-S. Chua. Scalable Detection of Partial Near-duplicate Videos by Visual-Temporal Consistency. *In ACM Multimedia, 2009.*

[6] H.-K. Tan, C.-W. Ngo and T.-S Chua. Efficient mining of multiple partial near-duplicate alignments by temporal network. *IEEE Trans. On circuits and systems for video Technology, 2010.*

[7] H.-T. Shen, J. Shao, Z. Huang, X.-F. Zhou. Effective and Efficient Query Processing for Video Subsequence Identification. *IEEE Trans. Knowl. Data Eng. 21(3): 321-334 ,March 2009.*

[8] http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.

[9] L.-f. Shang, L.-j. Y.-F. Wang, K.-P. Chan and X.-S. Hua. Real-time large scale near-duplicate web video retrieval. *In ACM Multimedia, 2010.*

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

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

[12] 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, 2009.*

[13] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. *In ACM Multimedia, 2008.*

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

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

[16] X. Wu, A. G. Hauptmann, and C.-W. Ngo. Practical elimination of near-duplicates from web video search. *In ACM Multimedia, 2007.*