International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.11 (2015), pp.339-354
http://dx.doi.org/10.14257/ijsip.2015.8.11.31

A Graph-Based Matching Algorithm on Sub-Sequence of Near Duplicated Video

Liao Ruihua and Zhao Kai

School of Information Science and Engineering, Hunan First Normal University, Changsha, 410205, China
Teaching Affairs Division, Pingdingshan University, Henan Pingdingshan, 467000, China
Lrhhnf@126.com

Abstract

As to solving the effectiveness and efficiency problem in the process of detecting the near duplicated video, we the thesis proposes a graph-based matching algorithm on sub-sequence of near duplicated video. The method will built similarity researching results based on the features of key frames feature into the query matching diagram, and then the near duplicated video detection is converted into a problem of searching the longest path in the matching results graph. As for its main advantage, firstly, it can find the best matching sequence in many cluttered matching results, which can effectively exclude a lot of noises brought by certain false "high similarity" matching, thus to some extent it can compensate the deficiencies of the underlying characterization force. Secondly, because it fully understands and uses the timing characteristics of the video sequences, the positioning accuracy of near duplicated video is with a high degree. Finally, multiple discrete paths existing in the matching results graphs are automatically detected, thus the situation where two video segments may exist several near duplicated videos can be detected once time. Experimental results show that the graph-based matching algorithm on video sub sequence improve the detection accuracy, at the same time improve the detection efficiency, which achieved good practice effect.

Keywords: *figure; near duplicated video; subsequence matching; key frames; path scatter*

1. Introduction

The important inner feature of video is its time characteristic, *i.e.*, the video is composed of successive video frames in the direction of the time, thus content-based video retrieval or near repeated video judgment usually need to be compared the similarities of the video frames set to judge the similarity between two videos, although, in principle, content-based near duplicated video detection is basically the same with the content-based video retrieval. However, in practical applications, there is a big difference in terms of implementation and existence. Content-based video retrieval pays more attention to the overall similarity of the query video and the goal video, that is, more of a "one to one" relationship, while for near duplicated video detection, its similarity retrieval may face more matches in the form. These matching forms can be summarized as "one-one", "one-many" and "many-many" [2]. These three forms are described in Figure 1. Because the existing forms of near repeated video are entirely dependent on the specific application of the target video. If the near duplicated video is operated by complex editing

ISSN: 2005-4254 IJSIP

and modification, its detecting work will be a typical matching problem of video sub-sequence. In the complex in the complex near duplicated video detection, the length of the target sub-sequence, the existing location and the existing frequency are unknown (the form of “many to many” showing in Figure 1 (c)); this form of matching task of video subsequence is much more complex than the ordinary video retrieval tasks.

Currently, the majority of detection methods on content-based near duplicated videos are mainly focused on copy detections of the two complete video [3] or the detections on partial near duplicated video [4]. Although these methods achieved good performance in their application scenarios, when faced with the complex situation of “many-many” in Figure 1, (c) these methods inevitably encounter detecting bottlenecks. In the detecting of near duplicated video based on content, on one hand, the spatial characteristics of video frames can be used to measure the similarity the video frames on the content of the video [5]. On the other hand, the time characteristic is an essential characteristic of the video. The near duplicated video detection is ultimately matched and positioned in the video frame sequence. In this framework, the similarity of the video spatial information is completed through consulting the similarity of the single frame image feature [6-7], but video subsequence matching methods in this study are focused on the combination between the video spatial information and time information to achieve the determination and position of near duplicated video subsequences. These are the problems of where the near duplicated video exists and how many times it exists. Typically, it is really hard to predict the appearing location and number of the near duplicated video on the target, therefore the obtained similarity matching results (*i.e.*, spatial information matches) based on the characteristics of the single frame image require an effective model to convert into a matching results in the time series. The match of search video and reference video at a certain point in the one-dimensional timeline can be described with a matching node, and the matching of a video sequence can be described with a bunch of points. This naturally reminds us the graph model can be used to solve the video subsequence matching problem. Graph transformation is widely used to describe complex structures with a natural and intuitive way [8].

There are many ways of the extraction methods on the video key frame [9, 13]. Due to different application scene of these methods, the focuses of its methods are different, therefore these methods have their inherent advantages, and on the other hand they are also inevitably exposing their shortcomings. For example, the inter-frame continuous comparison method [14] has the advantages of simple calculation, intuition, low computational complexity and so on, but the number of key frames is not controllable and there are limitations, such as the redundant information appearing in the same lens; while the method [15-16] based on clustering can use a common clustering algorithm and the key frame can reflect the global characteristics of the lens, *etc.*, but it cannot naturally use the timing characteristics of the video, which typically through a number of specific settings to ensure the next video frame can be assigned to the same class. Although the methods based on the object or event detection contains important semantic information, but they have strong correlation and dependence with the specific applications and such methods has high computational complexity.

This paper on the basis of summarizing the advantages and disadvantages of these methods proposes a graph-based video sub-sequence matching method. This method converts the similarity query results based on the video key frame feature into matching results graph, and furthermore change the near duplicated video detection into a problem of searching the longest path in the matching results graph. Specific work contents embodies in the following two aspects:

(a) Since the near duplicated video detection in this study has obvious relevance, namely testing two videos are the same or nearly the same, thus the method in this article does not focus on the structural analysis of the video, but on two main points: 1) effective elimination of redundant video frames to improve the detection efficiency; 2) maintaining the granularity of the extracted key frames to make sure the accuracy of test results. Based on these two objectives, the thesis proposes an automatic dual-threshold method to eliminate redundant video frame, divides the video into several video segments, and takes the methods of adopting fixed extracted frame rate on each video segments to extract three key frames to represent the video segment on each video. This method can effectively eliminate the redundant frames of video, but also ensure extracting the granularity of key frame, which achieves good practice effect.

(b) In order to eliminate the redundancy of the video on visual information and improve the matching speed of video sub-sequence, an efficient solution is to extract a set key frame which is far less than an entire video frame to represent the video content. The common practice of extracting the video key frames is to split the video camera, and then extract the key frames in visual content which is most representative of the lens from each lens. Currently, the key frame extraction method based on shot segmentation has many sophisticated algorithms [17-23], and in the field of video retrieval it also has a good application effect. But as to the application of near duplicated video detection, the method based on shot segmentation has limitations in the two aspects:

1) There is redundant visual information on the lens with similar content; 2) When the query video is very short (less than 5s), the granularity of key frame extraction is difficult to ensure the accuracy of detection. Therefore, the approach I picked up does not fully take the shot segmentation method to extract key frames, but rather follows the idea of dividing the video into the similar video clip on the visual content, and then extract fixing number of key frames on each video clip to represent the video clips, Figure 4 shows the distinction between the proposed video clips segmentation and commonly used shot segmentation method.

2. A Graph-based Matching Algorithm on Sub-sequence of Near Duplicated Video

A. Algorithm Framework

In order to facilitate the following discussion, this paper defines the related concepts as follows:

$Q = \{q_1, \ldots, q_{|Q|}\}$ and $R = \{r_1, \ldots, r_{|R|}\}$ are respectively represent the key frame collection of query video and reference video; q_i represents the i frame of the query video, and r_j means the j frame of the reference video; $|Q|$ is the number of video frames and $|R|$ represents the number of reference video frames.

$G = (N, E)$ means the matching results graph; N means the matching nodes collection; E means the set of edges.

$N = \{M_{i,j} \mid 1 \le i \le |Q|, 1 \le j \le |R|\}$. Node $M_{i,j}$ in the nodes collection represents the existing match which can satisfy specified constraints between the query video frames q_i and the reference video frame r_j. The most common used constraint is to result the similarity which is larger than a specified threshold T between the query video frames q_i and the reference video frame r_j. In addition, this also can be specified by $K - NN$, that is, the inquiry frame q_i returns to previous k nearest reference video frame.

$E = \{< M_{i,j}, M_{l,m} >\}$. $< M_{i,j}, M_{l,m} >$ means that there is a line between the node $M_{i,j}$ and the node $M_{l,m}$.

$sim(q_i, r_j)$ indicates the similarity between the query video frames q_i and the reference video frame r_j.

$P = \{p_1, \ldots, p_n\}$ shows the set of all the matching paths in the graph G.

$\vec{Q} = \{q_m, \ldots, q_n\}$, $1 \le m < n \le |Q|$ and $\vec{R} = \{r_m, \ldots, r_n\}$, $1 \le m < n \le |R|$ are respectively indicate the sub-sequence of query video and the sub-sequence of the reference video.

$sim(p_j)$ means the similarity between the sub-sequences of two corresponding video in the matching route.

$S_Q = \{a_1, \ldots, a_L\}$ and $S_R = \{b_1, \ldots, b_L\}$ are respectively represent the frame ID set corresponding to query video and the frame ID set corresponding to reference video in the matching route of the video subsequence, which means that in the matching path, the a_l frame of the query video and the b_l frame of the reference video are one-to-one correspondence; L represents the length of the matching path.

In this thesis, the author converts the matching results of the video in the space information into the matching results graph, and then changes the location problem of near duplicated video into the problem of searching the key route in the matching results graph. The specific flow is shown in graph 2. It's main steps include: 1) extraction of video key frames: 2) features matching based on key frames; 3) building the matching results graph based on matching results; 4) detection of matching route: 5) filtering matching route; 6) location of near duplicated sub-sequence.

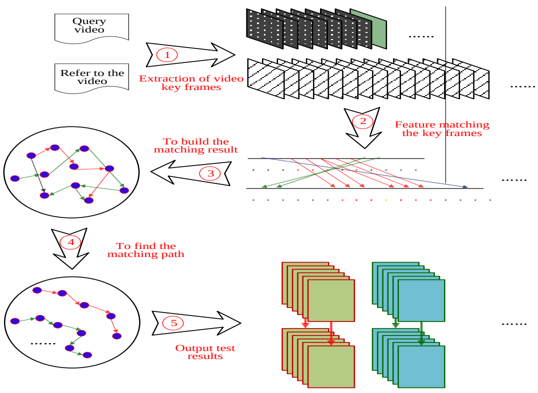


Figure 2. Flowchart of Video Subsequence Matching Based on Graph

The flowchart of video subsequence matching based on graph in Figure 2, and the key frame extraction is the first step based on the near duplicate video detection of the contents. The accuracy and granularity of extracting the video key frame are largely determines the quality of detecting the near duplicated video. Video is consisting of a sequence of frames with the timing characteristics, and when the human visual characteristics are taken into account, the similarity of adjacent video frames will be high. This makes the visual information in the successive video frames has a great redundancy in the time direction, and if during the detecting the near repeated video and extracting the features to each frame, the extra unnecessary calculations will be brought out. So the representative video key frames need to be extracted to reduce redundancy of visual information. Thus, when the near duplicated video is detected, there only needs to match the key frame sequence, thus the cost of accounting can be greatly reduced in the matching process.

According to the above definition, the matching task of video subsequence also can be defined as follows:

Video subsequence matching: in a predefined score function, the most similar part corresponding to the reference video sequence and the query video sequence is searched, that is, two video sub-sequences $\vec{Q}$ and $\vec{R}$ are searched to make highest similarity. In response to the abrupt and gradual of the video visual content, this paper adopts the dual-threshold method similar to the method proposed in literature [17]. There are two characteristics of this method as following: 1) usage of a dual-threshold method; one threshold T_h is adopted to detect the abrupt of frame visual information, and the other threshold T_l is adopted to detect the gradual of frame visual information. 2) The size of the two thresholds is not pre-determined, but according to the change of visual information of the video the size of threshold is automatically determined. Threshold T_h and threshold T_l are defined as follows: $T_h = \mu + \alpha\sigma$, $T_l = \beta T_h$. μ and σ are respectively mean and standard deviation of the entire video frame based on color features. According to the studies in literature [18], α is recommended to set between 5 and 6. According to the studies in literature [19], β is usually chosen from 0.1 to 0.5 .

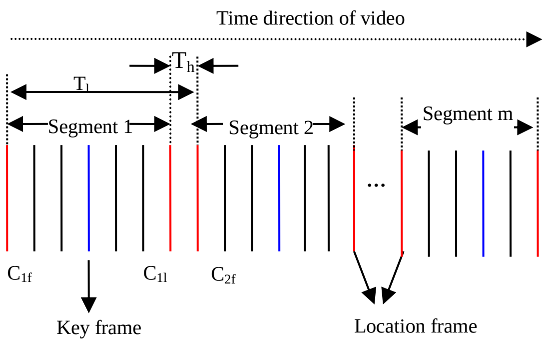
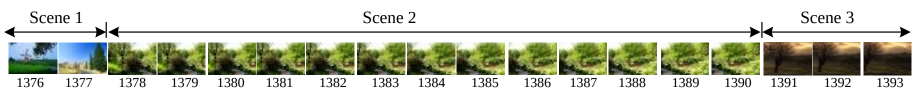


Figure 3. Elimination of the Video Redundancy Frame and Extraction of the Key Frame on f Adaptive Dual-Threshold

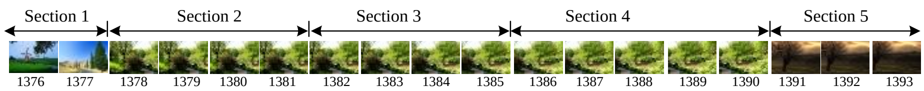
B. Algorithm Realization

The proposed theory of extracting key frames adopts the adaptive dual-threshold shown in Figure 3; in the figure there are successive video frames with a period of time, in which C_{1f} means the first frame in segment 1; C_{1l} means the last frame in segment 1; C_{2f} represents the first frame in segment 2; T_h means the mutation threshold; T_l means the gradient threshold. The segment 1 and segment 2 in the video must satisfy any one of the following two conditions:

1) The similarity between the last frame in the video segment 1 and the first frame in the video segment 2 $sim(C_{1l}, C_{2f}) < T_h$; the similarity between the first frame in the video segment 1 and the first frame in the video segment 2 $sim(C_{1f}, C_{2f}) < T_l$. The continuous video frames can be divided into several visually similar video segments through the adaptive dual-threshold segmentation method, and three frames are extracted to represent each video segment. These three frames can be divided into the first frame, the key frame and the last frame (shown in Figure 3), in which the key frame and the average frame (the means of all the frame features in this segment) can be replaced by the most similar frame. Key frame is used to match a video sequence, while the first frame and the last frame are mainly used for precise positioning and auxiliary matching, which allocates contiguous ID to the divided video segments with the time direction. Figure 4 illustrates the difference between the method of proposed adaptive dual-threshold method eliminating the redundant video frames and the method of ordinary shot-based segmentation. Figure 4 (a) is the result obtained by the method using the shot segmentation; Figure 4 (b) segmentation results obtained by using the automatic dual-threshold to eliminate redundant video frames; in Figure 4, the distance (the distance between features based on the color) between the last frame (ID 1377) in the video segment 1 and the first frame (ID 1378) in the video segment 2 is greater than the mutation threshold T_h; the distance between the first frame (ID 1378) in the video segment 2 and the first frame (ID 1385) in the video segment 3 is greater than the mutation threshold T_l. Therefore, based on the proposed adaptive dual-threshold segmentation method, the lens 2 can be further divided segment 2 and segment 3, which has a smaller particle size. Therefore, it is called as sub-lens video or video segment in this article.



(a) Results of Shot Segmentation



(b) Segmentation Results of Adaptive Dual-Threshold

Figure 4. Results Contrast of Adaptive Dual-Threshold Segmentation and the Lens Segmentation

After the matching results figure is built, the goal of matching the near duplicated video subsequence is to find the longest matching path in the Figure. So far, the matching problem of near duplicated video subsequence is converted into the problem of searching longest path in the matching results graph. Many classical algorithms can be used to find the shortest path of starting point of each node, such as Dijkstra, Bell man Ford, Floyd warshall and so on. Since what this thesis tying to find out is the longest path between any two nodes in graph, and thus the Floyd warshall algorithms is chosen. By using the *Floyd_warshall* algorithm, the longest path between any two points in the graph can be found out one time, and these paths determine the position of near duplicated video, but also determine the length of, and avoid exhaustive method to determine the position and length of near duplicated video, in which the detection efficiency is effectively improved.

As to any node in the figure, there is no path which is start with it, but maybe there is a path or several paths. node $M_{1,29}$,node $M_{1,76}$,node $M_{2,76}$ in figure have no path (the path is the node itself); while for node $M_{1,26}$, there are 16 paths. Wherein there are four longest path (L = 6), namely:

Route1: $M_{1,26} \rightarrow M_{1,27} \rightarrow M_{1,28} \rightarrow M_{1,29} \rightarrow M_{1,30} \rightarrow M_{1,31} \rightarrow M_{1,32}$;

Route 2: $M_{1,26} \rightarrow M_{1,27} \rightarrow M_{1,28} \rightarrow M_{1,29} \rightarrow M_{1,30} \rightarrow M_{1,31} \rightarrow M_{1,33}$;

Route 3: $M_{1,26} \rightarrow M_{1,27} \rightarrow M_{1,28} \rightarrow M_{1,29} \rightarrow M_{1,30} \rightarrow M_{1,32} \rightarrow M_{1,33}$;

Route 4: $M_{1,26} \rightarrow M_{1,27} \rightarrow M_{1,28} \rightarrow M_{1,29} \rightarrow M_{1,30} \rightarrow M_{1,32} \rightarrow M_{1,34}$;

These 16 matching paths are stacked in the time sequence, so when the final matching sequence is outputted, these laminated paths on times must be merged. Then according to the timestamp information corresponding to query video and the reference video of the matching paths' starting and final node, the location of these two near duplicated video is located. For the merging of these cascading paths, this thesis takes the following matching strategy to the longest path:

The matching strategy of the longest path: Given any two paths p_i and p_j , let's set the beginning point and final point of p_i respectively are $M_{a,b}$ and $M_{c,d}$. Then there are three possibilities of the two paths in the timing position relationship: divided, contained and intersected. The combined path is also divided into three cases:

1) Divided: if $c \le e$ and $d \le f$, or $g \le a$ and $h \le b$, then the path p_i is away from path p_j . At this point there is no lamination of these two paths in the video sequence, so the path p_i and path p_j are two mutually independent paths, and the combined paths are still two separate path p_i and path p_j .

2) Contained: if $a \le e$, $b \le f$, $c \le g$ and $d \le h$, the path p_j contains the path p_i and the combined path is p_j . Conversely, if $a \ge e$, $b \ge f$, $c \ge g$ and $d \ge h$, the

path p_i contains the path p_j and the combined path is p_i.

3) Intersected: If $e < a < g$, $f < b < h$, $c > g$ and $d > h$, the starting point $M_{a,b}$ of the path p_i is contained in the path p_j, while the final point $M_{c,d}$ of p_i falls after the end point $M_{g,h}$ in the path p_j, so the combined path is $p_j \to M_{c,d}$, *i.e.*, node $M_{g,h}$ which is added after p_i is chosen as end point.

If $a < e < c$, $b < f < d$, $g > c$ and $h > d$, the starting point $M_{e,f}$ of the path p_j is contained in the path p_i, while the final point $M_{g,h}$ of p_j falls after the end point $M_{c,d}$ in the path p_i, so the combined path is $p_i \to M_{g,h}$, *i.e.*, node $M_{g,h}$ which is added after p_i is chosen as end point.

After the matching path in the graph is merged through the longest path matching strategy, the discrete paths are ultimately got in the matching results figure, that is, there is no stacked path in time. For example, as for all paths in the matching results figure shown in Figure 6, when the longest path matching strategy is merged, the final matching path is as follows:

$$M_{1,26} \to M_{1,27} \to M_{1,28} \to M_{1,29} \to M_{1,30} \to M_{1,32} \to M_{1,34}$$

According to the final matching path, the frame set got in the query video sequence is $S_Q = \{1,2,3,4,5,7,8\}$ and the frame set got in the reference video sequence is $S_R = \{26,27,28,29,30,32,34\}$. Finally, according to the time-stamp information of the starting frame in S_Q and S_R, the position of the two videos sequences are located corresponding to the location in the video. Such as according to the corresponding time point of the first frame and the eighth frame in the query video, the location of the sub-sequence can be located in the in query sequence, and according to the corresponding time point of the 26th frame and the 34th frame in the reference video. Obviously, the proposed graph-based method can directly detect the situation of several near duplicated video presence existing in these two videos. Furthermore, in order to filter the matching noise brought by some short paths (such as some paths with the length as 2 or 3). In this paper, through the experimental study, Equation 2 can be used to measure the similarity between the sequences of two video:

$$sim(p_i) = \frac{\sum_{k=1}^{L} sim_k(M_{i,j})}{L} \log(1+L) \tag{1}$$

Wherein, L is the number of nodes in the path; $M_{i,j}$ is a node in the path; $sim(M_{i,j}) = sim(q_i, r_j)$.

3. Experiments and Analysis

A. Experimental Procedure

The experiment shown in Figure 4 is taken to process experiment which takes lens divided data as an example. The following matching results data is obtained.

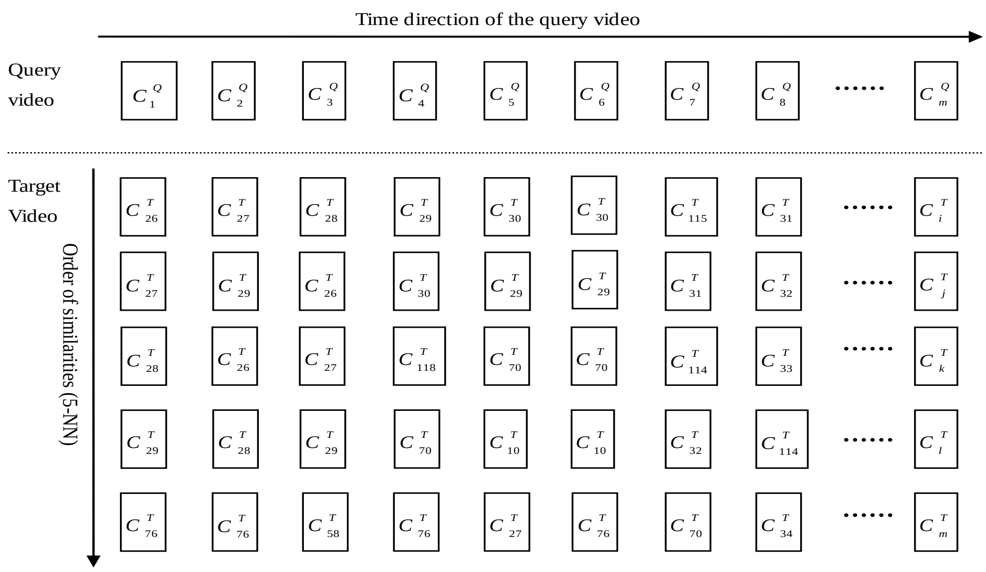


Figure 5： matching results sample of the query video and reference video

Illustration： The first eight rows dates shown in the figure are taking the matching results in these two videos

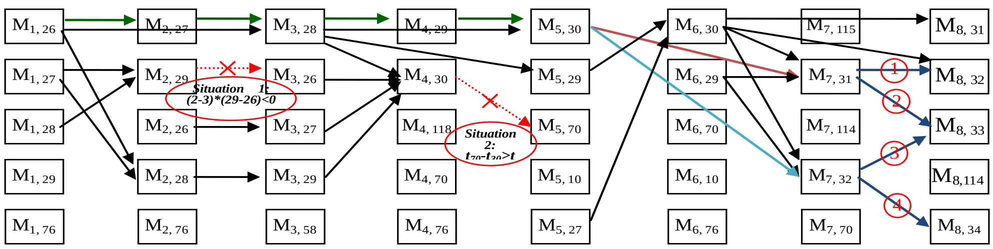


Figure 6. Building the Matching Results Graph (the Matching Results Come From Figure 5)

According to the matching process shown in Figure 2, the preliminary work to construct the matching results figure also includes: a) extracting video key frame feature; 2) consulting the similarity between the key frame features of the query video and key frame characteristics of the reference video. These two studies can be seen in the author’s research papers [31-32]. The building of matching results graph is based on the similarity consulting results between the key frame features of the query video and key frame characteristics of the reference video.

As to the key frame set $Q = \{q_1, \dots, q_{|Q|}\}$ of query video and the key frame set $R = \{r_1, \dots, r_{|R|}\}$ of reference video, the similarity $sim(q_i, r_j)$ between query video frame q_i and reference video frame r_j is calculated; as to each key frame q_i in the query video, according to the size of similarity value it is returned to the k most similar key frame in the reference frame.

The size of k is identified by the number of the segments of the divided video, *i.e.*, $k = \alpha n$; α is a scaling factor, which typically is 0.05. The matching method effectively avoids the returned defects of needing a rigid threshold value to determine the matching results. Figure 5 is a schematic diagram of matches.

The data in first eight rows in Figure 8 are matching results derived from the first two real videos. When the data is matching, each frame q_i in the set of query video $Q = \{q_1, \dots, q_{|Q|}\}$ is ordered by time, and each frame r_j in the set of reference video $R = \{r_1, \dots, r_{|R|}\}$ is taken the similarity retrieval. Take the first column in figure for an example, there is $sim(q_1, r_{26}) \geq sim(q_1, r_{27}) \geq sim(q_1, r_{28}) \dots$. The first k returning results is taken, and in the figure $k = 5$. Each match in Figure 5. can be converted the node $M_{i,j}$ in the matching results figure. After the figure node is obtained, the next step is to determine whether there is directed edge between the two nodes, and in order to describe whether two nodes are reachable in the figure this paper makes two definitions:

1) Time direction consistency of the nodes. As to node $M_{i,j}$ and node $M_{l,m}$, if $(i - l) * (j - m) > 0$, the two nodes have consistency in a time direction.
2) The time jump degree of node. The time jump degree between node $M_{i,j}$ and node $M_{l,m}$ can be defined as:

$$\Delta t_{l,m}^{i,j} = \max(|t_i - t_l|, |t_j - t_m|) \tag{2}$$

B. Experimental Results

If there is an edge between two nodes in the matching figure, there are two conditions that must be simultaneously meted as following:

(1) The two nodes must be satisfied the consistency in the time direction;

(2) The time jump degree of two nodes is $\Delta t < \tau$ (τ is threshold of time jump degree)

The condition (1) represents the time direction of the query video represented by two nodes is consistent with the time direction of the target video graph, which may seems reasonable to do so, because the video sequence is a time s sequence and the time direction of the coping video and the copied video is the same. The direction of increasing time is directed edges between nodes direction. The condition (2) represents the jumps of the matching results represented by two nodes in the time direction can not exceed a certain threshold, otherwise there is no correlation between the two matching results. According to the above methods and conditions, the matching results in Figure 5 can be converted into a matching figure, and there is obviously a directed acyclic graph in the matching results figure. The matching results of the eight rows in Figure 5 can be converted into the matching results shown in Figure 6. The author also had a comparative experiment to obtain the most optimal time jump threshold as shown in Figure 7; T1-T10 in the figure are represent 10 different types of copied video (For details, see [1]); the experimental data shows that a relatively appropriate time jump threshold should be between 10 to 29 seconds. On the other hand, the author also makes the statistics to the time length of the obtained video clips of the adaptive dual threshold method as shown in Figure 8. The time length of video segments shown by data in the figure mostly concentrates between 10 and 30 seconds, which also verified the reasonability of choosing threshold of time jump degree between 10 and 30 seconds.

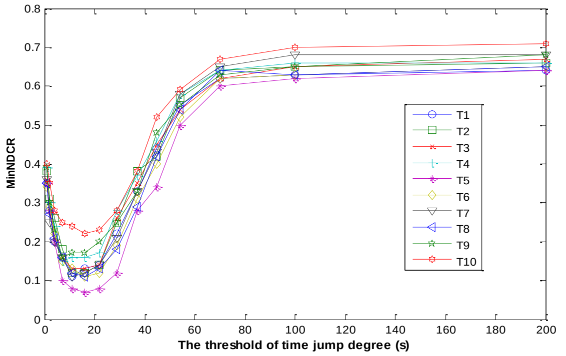


Figure 7. Selection of Threshold of Time Jump Degree

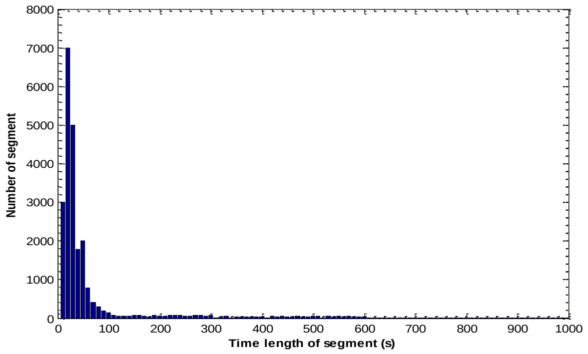


Figure 8. Distribution of Time Length of Segments after the Video is Divided by Adaptive Dual Threshold Method

C. Experimental Analyses

The matching results based on the video frame of visual characteristics do not include the time characteristics of the video, while the graph-based video sub sequence matching method rearranges and refines the matching results in the time sequence based on video spatial characteristics, which effectively integrate the spatial characteristics and timing characteristics of the video. This method has the following main advantages:

1) The optimal matching sub-sequences can be found from the matching results with the disordered video space feature. Figure 9 uses the challenging copy type: "PIP" copy type, which is used to illustrate the advantages of graph-based approach. Dashed frame A contains some queries video frames in time direction. Dashed frame B is the reference video frame collection, in which the query video frames set (the similarity of the reference video frame ranged from highest to lowest) is returned based on similarity. Then, graph-based subsequence matching method can automatically find an optimal matching path in the matching results. In Figure 9, the path indicated with the red arrow is the obtained final matching result of two videos.

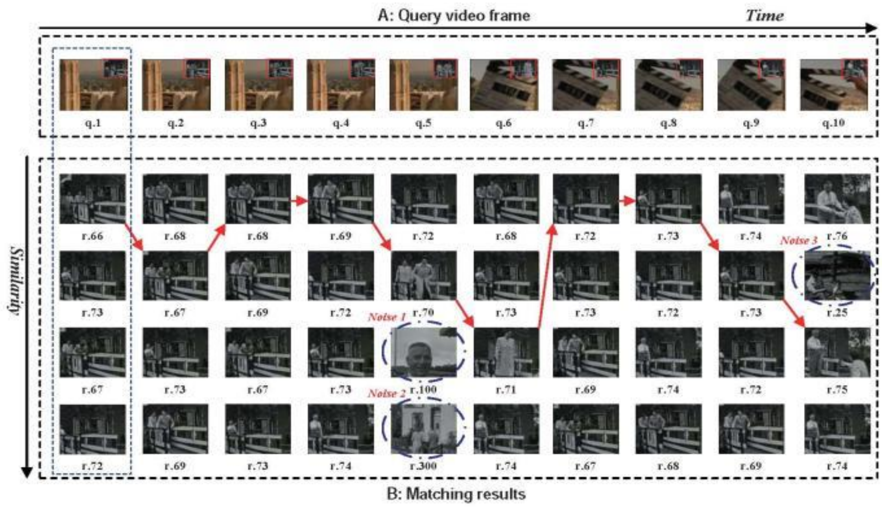


Figure 9. Method in the Figure Can Automatically Find the Optimal Path in the Matching Results

2) The noises of matching results in video spatial characteristics can be automatically removed some of. In the detection process, the matching results based on the spatial characteristics always make some "noise" (as noise 1-3 shown in Figure 9). Graph-based subsequence matching method can use consistency in time direction and the threshold to remove these noises.

3) The change of adaptive video frame rate can be. Query video frame rate in Figure 10 is 2 times of query video frame rate in Figure 9. Experimental results show that the graph-based matching method can effectively cope with different copies of the video frame rate.

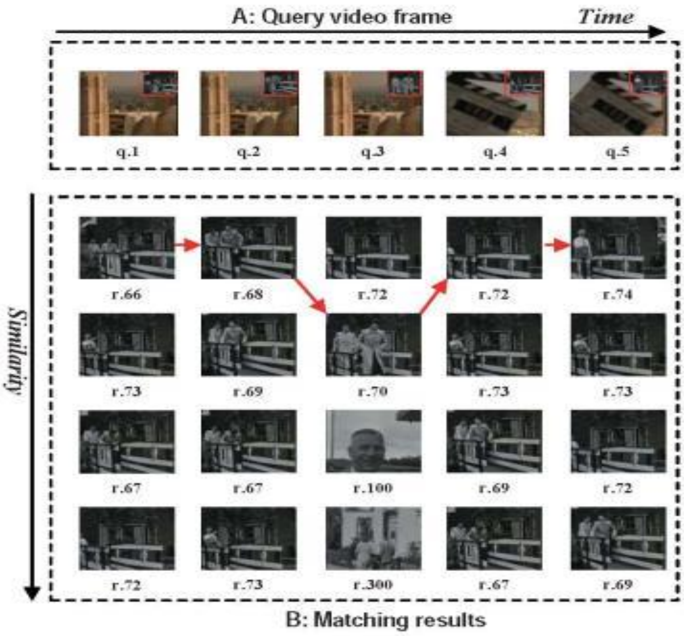


Figure 10. Changes of Adaptive Video Frame Rate Method in the Figure

4) Two video which can be simultaneously detected contain several near duplicated video segments. In practice, the near duplicated video segments may appear multiple times in a video. For example, Figure 11 shows the same advertisement is broadcast several times in different time. The experimental results figure 12 shows that: the matching method of video sub sequence based on graph can detect several near duplicated video segments contained in the video one time. At the same time, it is also of practical value, for example, the advertisers want to know whether the television broadcast a designated ads contracted in the time and frequency; the approaches based on graph can be used to make automatic detection, and artificial watch video way to confirm is not needed.

Figure 11. The Same Advertisement is Broadcast at Different Time Points

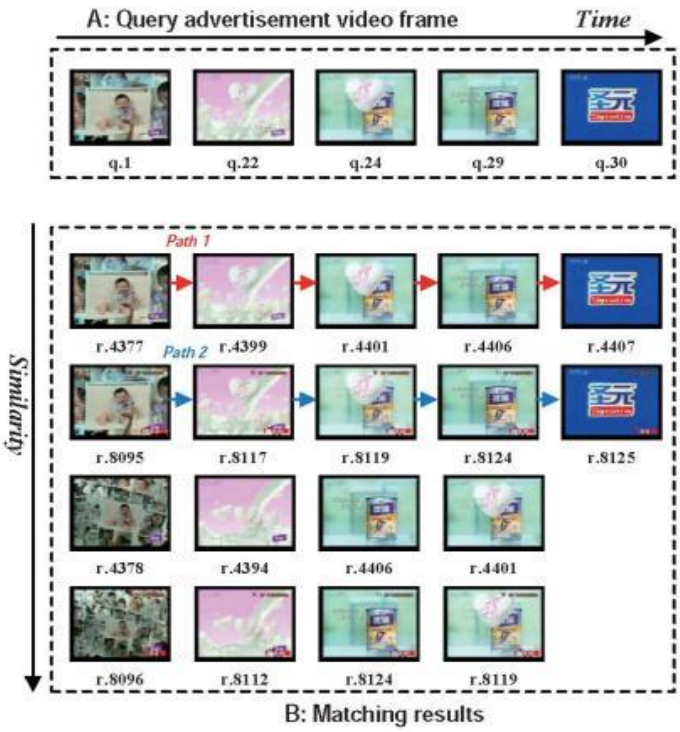


Figure 12. Figure Method can Simultaneously Several Near Duplicated Video Segments Obtained in the Two Videos. Path 1 and Path 2 Represent the Test Results of the Same Advertisement Broadcasting in Different times

4. Conclusions

Because there are lots of uncertain factors existing in process of near duplicated video detection, such as whether exist near duplicated video in the target vide, how long the near duplicated video segments are and what position the near duplicated video locates, which makes the detection of near duplicated is harder compared with the usual video retrieval. Currently, there are two main drawbacks of the subsequence matching algorithm for near duplicated video detection: Firstly, a similar threshold is needed to determine in the matching process to decide the returned results, while for different videos, it is really difficult to determine a common threshold. Secondly, it is cost a lot to detect the copied video with all possible lengths and possible positions by using exhaustive. So this thesis proposes a matching algorithm of video sub-sequence based on graph. This method skillfully converts video sequence matching results into the matching results graph, and then converts the video copy detection into a problem of finding the longest path in the matching results figure. There are several distinct advantages of this method based on graph: (1) a graph-based approach can find out the most optimal matching sequence in a number of clutter, which effectively eliminate some noises brought by some false "high similarity" matches, so it can make up inefficient defects of describing the underlying characterization. (2) Because it makes full consideration and use of the timing characteristics of the video sequences, the positioning accuracy of near duplicated video is really high. (3) Graph-based sequence matching method can automatically detect the plurality of discrete paths, so it can locate in the batch several near duplicated video in these two videos. (4) Compared with the exhaustive testing, this method has obvious advantage on the detection speed.

Acknowledgments

This Work was supported by the Soft science research project of Hunan province key projects (NO. 2013 zk2014).

References

[1] Xu, Z. Y. Feng and Y. Z. Li, “Fair Channel allocation and power control for uplink and downlink cognitive radio networks”, IEEE., Workshop on mobile computing and emerging communication networks, **(2011)**, pp. 591-596.

[2] W. Q. Yao, Y. Wang and T. Wang, “Joint optimization for downlink resource allocation in cognitive radio cellular networks”, IEEE., 8th Annual IEEE consumer communications and networking conference, **(2011)**, pp. 664-668.

[3] S. H. Tang, M. C. Chen and Y. S. Sun, “A spectral efficient and fair user-centric spectrum allocation approach for downlink transmissions”, IEEE., Globecom., **(2011)**, pp. 1-6.

[4] L. Sun, X. N. Zhu and Z. M. Zeng, “Downlink power control in cognitive femtocell networks”, IEEE., International conference on wireless communications and signal processing, **(2011)**, pp. 1-5.

[5] M. J. Mirza and N. Anjum, “Association of Moving Objects across Visual Sensor Networks”, Journal of Multimedia, vol. 7, no. 1 **(2012)**, pp. 2-8.

[6] K. Suhairi and F. L. Gaol, “The Measurement of Optimization Performance of Managed Service Division with ITIL Framework using Statistical Process Control”, Journal of Networks, vol. 8, no. 3 **(2013)**, pp. 518-529.

[7] G. Yan, Z. Yue-Fei, G. Chun-Xiang, F. Jin-long and H. Xin-Zheng, “A Framework for Automated Security Proof and its Application to OAEP”, Journal of Networks, vol. 8, no. 3 **(2013)**, pp. 552-558.

[8] R. Berangi, S. Saleem and M. Faulkner, “TDD cognitive radio femtocell network (CRFN) operation in FDD downlink spectrum”, IEEE, 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, **(2011)**, pp. 482-486.

[9] Pearson S., “Taking account of privacy when designing cloud computing services”, In CLOUD '09: Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing, IEEE Computer Society, Washington, DC, USA, **(2009)**, pp. 44–52.

[10] http://dx.doi.org/10.1109/CLOUD.2009.5071532

[11] Y. Yona and M. Feder, "Efficient parametric decoder of low-density lattice codes", IEEE International Symposium on Information Theory: June 28-July 3, 2009, Seoul, Korea. New York, NY, USA: IEEE, vol. 8, **(2009)**, pp. 744-748.

[12] B. Kurkoski and J. Dauwels, "Message-passing decoding of lattices using Gaussian mixtures", IEEE International Symposium on Information Theory: June 6-11, 2008, Toronto, Canada. New York, NY, USA: IEEE, vol. 8, **(2008)**, pp. 2489-2493.

[13] BICKSON D, IHLER A and AVISSAR H, “A low-density lattice decoder via non-parametric belief propagation”, Forty-Seventh Annual Allerton Conference on Communication, Control and Computing: Sep 30-Oct 2, 2009, Illinois, USA. Monticello, IL, USA: IEEE, vol. 1, **(2010)**, pp. 439-446.

[14] Mondol, J.-A.M., “Cloud security solutions using FPGA. In Communications”, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on, **(2011)**, pp. 747-752.

[15] Lv, Zhihan, Lb Feng, H Li, and Sz Feng, "Hand-free motion interaction on Google Glass", In SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications, ACM, **(2014)**, p. 21.

[16] Zhong, Chen, S. M. Arisona, Xf Huang, M. Batty, and G. Schmitt, "Detecting the dynamics of urban structure through spatial network analysis", International Journal of Geographical Information Science 28, no. 11 **(2014)**, pp. 2178-2199.

[17] Li, Wubin, J. Tordsson, and E. Elmroth, "An aspect-oriented approach to consistency-preserving caching and compression of web service response messages", In Web Services (ICWS), 2010 IEEE International Conference on, IEEE, **(2010)**, pp. 526-533. L. L. yuan and L. C. lin, "A multicast routing protocol with multiple QoS constraints", Journal of Software, vol. 15, no. 2, **(2004)**, pp. 286-291.

[18] Q. He, C. Han, "Satellite Constellation Design with Adaptively Continuous Ant System Algorithm", Chinese Journal of Aeronautics. vol. 6, no. 4, **(2007)**, pp. 297-303.

[19] L. Y. Ren, "Study on Scheduling Optimization in Construction Project of Lagerstroemia Hope City", Xi'an University of architecture & techinology, vol. 6, no. 2, **(2011)**, pp. 12-17.
[20] B Yu, Z Z Yang and B Z Yao, "A hybrid algorithm for vehicle routing problem with time windows", Expert Systems with Applications, vol. 38, no. 1, **(2011)**, pp. 435-441.
[21] S. Kudekar, T. Richardson and R. Urbanke, "Threshold saturation via spatial coupling: why concolutional LDPC ensembles perform so well over BER", IEEE Transactions on Information Theory, **(2011)**, vol. 57, no. 2, pp. 803-834.
[22] Y. Geng, J. Chen and K. Pahlavan, “Motion detection using RF signals for the first responder in emergency operations: A PHASER project”, 2013 IEEE 24nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), London, Britain, **(2013)** September.
[23] Y. Geng, J. He and K. Pahlavan, “Modeling the Effect of Human Body on TOA Based Indoor Human Tracking[J]”, International Journal of Wireless Information Networks, vol. 20, no. 4, pp. 306-317.
[24] Lv, Zhihan, Lb Feng, H. Li, and Sz Feng, "Hand-free motion interaction on Google Glass", In SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications, ACM, **(2014)**, p. 21.
[25] Zhong, Chen, S. M. Arisona, Xf Huang, M. Batty, and G. Schmitt, "Detecting the dynamics of urban structure through spatial network analysis", International Journal of Geographical Information Science 28, no. 11, **(2014)**, pp. 2178-2199.
[26] Li, Wubin, J. Tordsson, and E. Elmroth, "An aspect-oriented approach to consistency-preserving caching and compression of web service response messages", In Web Services (ICWS), 2010 IEEE International Conference on, IEEE, **(2010)**, pp. 526-533.

Author

Liao Ruihua, female, was born in HuNan, China, in 1970. Now, she is a associate professor. Her main research area is soft engineering and image processing.

