Large-scale Video Copy Retrieval with Temporal-concentration SIFT

Yingying Zhu¹, Xiaoyan Huang², Qiang Huang¹, Qi Tian^{3,*}

¹*School of Computer Science and Software Engineering, Shenzhen University, 518060, China*

²*Oracle Research and Development Center Shenzhen Co., Ltd, 518057, China*

³*Department of Computer Science, University of Texas at San Antonio, TX78249, USA*

Abstract

The scale-invariant feature transform (SIFT) feature plays a very important role in multimedia content analysis, such as near-duplicate image and video retrieval. However, the storage and query costs of SIFT become unbearable for large-scale databases. In this paper, SIFT features are robustly encoded with temporal information by tracking the SIFT to generate temporal-concentration SIFT (TCSIFT), which highly compresses the quantity of local features to reduce visual redundancy, and keeps the advantages of SIFT as much as possible at the same time. On the basis of TCSIFT, a novel framework for large-scale video copy retrieval is proposed in which the processes of retrieval and validation are implemented at the feature and frame level. Experimental results for two different datasets, i.e., CC_WEB_VIDEO and TRECVID, demonstrate that our method can yield comparable accuracy, compact storage size, and more efficient execution time, as well as adapt to various video transformations.

Keywords: Video copy retrieval; SIFT; Spatio-temporal features; Frame validation

1. Introduction

With the development of internet technology and the popularity of multimedia devices, video data are now publicly available and continue to grow rapidly. Users shoot a large number of videos and upload them to the internet. Due to the mass videos on the internet, content-based video retrieval (CBVR) has become a hot topic of intensive studies. CBVR schemes always use the query by example paradigm [1-3]. However, some users download these videos and edit them easily and then upload these videos again. A large sum of video copies have gushed out, leading to piracy problem, content protection and so on. Content-Based Video Copy Detection (CBVCD)[4,5] has since been proposed to detect whether a query sequence is a copied version of a reference sequence. Instead of being an identical or near-replicated video sequence, a copy may be a transformed video sequence [6]. As a result, a copy may be visually less similar. Most existing CBVR algorithms address the similar retrieval problem, and the CBVCD algorithms focus on detecting the copy relationship between a pair of videos. In the context of copy detection, a CBVR system may give rise to high false alarm rates [7] which is undesirable. On the other hand, unlike a CBVR system, response time is not too critical for a CBVCD system. The excessive number of combinations between videos when performing detection, so the time consumed by the detection is relatively long, compared to CBVR.

Recently, Content-Based Video Copy Retrieval (CBVCR) has been proposed to solve the problem mentioned above and therefore generated a great deal of interest in the research community and the multimedia industry. CBVCR has merged with CBVR and CBVCD, which refers to the situation where a video database is established, and the user inputs a query video into the search interface. The system receives the query video, processes it into features and then signatures, finally it matches the query signatures with those in the database[8]. It aims at retrieving in a database all of the modified versions or previous versions of a given video clip. Yuan et al.[9] gave a complete list of all query-by-video-clip methods developed in the last decade. From these methods, we can learn that CBVCR usually focuses on robust stable feature extraction from videos. Features (or descriptors), extracted from images and videos, consist of extraction characteristics describing important information in the images and videos.

* Corresponding Author
Email: qitian@cs.utsa.edu

Video features used for CBVCR can be roughly divided into three categories: visual features (color, texture, shape, contour, and some local invariant features, etc.) [10-12]; motion features (camera motion, object motion, person action, etc.) [13-15]; and spatio-temporal features, which play an important role in the research of CBVCR. A color histogram is a representation of global features that is compact, computationally efficient, and often employed to detect near-duplicate videos [16,17]. However, it contains no geometric, shape, or texture information and is sensitive to color changes. Similar histograms may be generated from very different images if they have similar color patterns. [18-21] use local detectors and descriptors which can capture object-level information and are invariant to scale and affine transformations. However, it is computationally expensive to match a large number of local features for image similarity computation. Coskun et al. [22] extracted video features from the time and spatial domain. They considered the sequence of video frames as a 3-dimensional matrix, in which they extracted DCT and RBT as the global spatio-temporal features. Mani et al. [23] extracted the temporally informative representative images (TIRI) which are weighted sum images of a sequence of frames, and then calculated DCT as the spatio-temporal features to detect the video copy. Both [22] and [23] considered video features from the spatio-temporal domain, but they extracted several transform coefficients as the global features, which are too highly compressed and not robust and distinctive enough for some video transformations. Guo et al. [24] tracked local interesting points through adjacent frames and used them as the matching features in copy detection. This paper proposed encoding the coordinate offset between the adjacent interesting points and then calculated the 5 bins histogram to describe the track. Wu et al. [25] presented the spatio-temporal trajectory as the detecting feature. The theory of spatio-temporal trajectory was similar to [24] in that it encoded the coordinate offset of the adjacent interesting points and quantified an L-length track to L-1 states, finally consummating in the building of a Markov motion model with these states to obtain the Markov stationary distribution trajectory motion features. [24] and [25] extracted the motion features of interesting points. However, these features are unstable when a video is copied done using the transformations such as: translation, flip, distortion, camcording and so on. On the other hand, in some videos, the screen will stay for long time leading to the motion features of interesting points being indistinctive. Geert Willems et al. [26] first presented spatio-temporal interest points that were at the same time scale-invariant (both spatially and temporally) and that densely covered the video content. However, this is suitable for action recognition instead of whole scene detection. Law-To et al. [27] employed a Harris detector to locate all keypoints. A Harris detector can generate a relatively smaller feature set, making it feasible to process with the proposed method within reasonable time and with reasonable space. R. Roopalakshmi [28,29] integrated visual-acoustic fingerprints based on visual features (color feature or SURF) and audio features (MFCC or spectral centroid) to improve the copy detection performance, but it was time-consuming and computing-intensive to handle the resultant multimodal signatures. These methods are still not robust enough to be applied to a large-scale database. Although a few algorithms [30-32] have been developed to retrieve a large-scale video database, their low efficiency limits their potential use on a web-scale dataset.

Thus, it is important to carefully decide which representation to adopt for CBVCR. Ideally, the representation should: (i) discriminate different transformations well; (ii) be easily extracted from the raw images/videos to allow for fast processing; and (iii) lie in a low dimensional space or small quantity to avoid a computational complexity. Naturally, it is easy to consider that SIFT features are relatively insensitive to changes in viewpoint and considerably stable among different changes of an image, such as scale, orientation, lighting conditions, etc. However, the high dimensionality and large quantity of SIFT, especially for videos, lead to enormous storage and computation costs. In this paper, we propose a novel methodology for the compact representation of a video's spatio-temporal structure by utilizing the unique temporal information of a video to compact a large number of video SIFTs, and keeping the advantages of SIFT as much as possible at the same time. In comparison with all of the existing research as surveyed above, our proposed algorithm and framework can achieve the following advantages: (i) we can track the local features through video to generate the temporal-concentration SIFT (TCSIFT), which can highly compress the quantity of local features to reduce visual redundancy; (ii) SIFT features can be robustly encoded with this temporal information, which is used effectively in measuring similarity and can represent SIFT features compactly; and (iii) we propose a novel framework that monitors a continuous stream with a large video dataset. The framework consists of two matching paradigms: one is at the feature level and the other is at the frame level, at which the processes of retrieval and validation are improved for large-scale datasets.

The remainder of this paper is organized as follows. Section II gives an overview of the proposed video copy retrieval framework. Section III provides a detailed description of how to track SIFT descriptors to generate TCSIFT. Section IV presents the way to retrieve and validate at the feature level and frame level. In Section V, a few experiments are conducted to show the performance of our framework and algorithm and some comparisons are

made. A conclusion will be presented in Section VI.

2. An Overview of the Proposed Video Copy Retrieval Framework

In this section, we present a framework to address the problem of content-based copy clip retrieval of any type of videos for large-scale datasets. This section will give the copy definition, the issues addressed in this paper and an overview of the proposed framework.

In [33], Alexis et al. gave the Copy Definition: A video O_1 is a copy of a video O , if $O_1 = t(O)$, $t \in T$, where T is a set of tolerated transformations. O is called the original video. Note that T can contain combinations of server transformations and that a video $O_3 = t_3 \circ t_2 \circ t_1(O)$, $t_1 \in T$, $t_2 \in T$, $t_3 \in T$ can be a copy of O as well as $O_1 = t(O)$ and $O_2 = t_2 \circ t_1(O)$. Five types of transformation examples are shown in Table 1.

An overview of our framework is shown in Figure 1. The framework consists of two parts: offline part and online part. The details will be discussed in sections 3~5.

Table 1. Examples of five video transformations

Camcording	Picture_in_picture	Photometric	Editing	Formatting

Fig.1. The framework diagram of video copy retrieval

In the offline part, a feature database would be built for the video database, in which the retrieval process is done, and an index is constructed to help organize the features efficiently and effectively for retrieval. Firstly, each video in the database is represented by a number of representative frames which are sampled from the video uniformly. Secondly, to build the feature database, spatio-temporal interesting points (TCSIFT) are extracted from these representative frames which are regarded as the basic query cell. Finally, the locality sensitive hash (LSH) indexing [34] which is better suited for large-scale databases is used to organize all TCSIFTs in the feature database.

In the online part, the query video do in the same way as the video in the database to get TCSIFTs, and then similar TCSIFTs are retrieved from the feature database using the LSH method. The matched frames can be derived from these similar TCSIFTs. Finally, the videos containing a number of matched frames are regarded as the final results and the returned videos will be re-ranked according to their content similarity to the query video clip and their time consistency.

3. Temporal-concentration SIFT

3.1. Representative frames extraction

Due to the temporal redundancy of video, there is no demand to process all frames of the query clip or the database videos, which would be time-consuming. The popular method is shot-keyframe. Most existing shot-keyframe methods have the following problems: i) shot segmentation may be different between the query video clip and database video, resulting in the different key frames that cannot match each other during the searching process; ii) video transformations, noises and camera motion can be obstacles. In these cases, many of the shot segmentation algorithms cannot achieve high segmentation reproducibility; iii) one or several key frames often cannot represent the whole or most of the shot content, so the shot-keyframe method may risk for losing much information; and iv) a video is an object in both the temporal and spatial domain; the shot-keyframe method only maintains a part of the spatial information but abandons the temporal information.

In this paper, a uniform sampling strategy is adopted to extract the frames, called representative frames, and different sample rates are employed for the query video and video database. To achieve a high repeatable rate between the videos in the query set and video database, the sample rates between them should meet the following equation:

$$f_q = n \times f_r (n \in \mathbf{Z}, n \geq 1) \quad (1)$$

where f_q and f_r are the sample rates for the query video and the reference videos, respectively. They are set to be 1 frame per second and 0.5 frames per second respectively, namely, $f_q = 2 \times f_r$. $\mathbf{Z}$ is the integers set.

3.2. Spatio-temporal feature extraction

The number of sift features of each frame ranges from a few dozen to more than a thousand, differing according to the size and visual content of video frames. Because the number of SIFT features of a video is incredible, the retrieval problem is considered large-scale. The extracted sift features have much redundancy and weak discrimination because of video redundancy in temporal domain. This is particularly prominent in some interviewing videos or story videos. Moreover, SIFT features can only represent the image spatial information, and cannot reflect the temporal information of videos.

Inspired by [35], we proposed a novel algorithm to solve these problems by replacing SIFT with temporal-concentration SIFT (TCSIFT). TCSIFT consists of two parts: temporal trajectory and descriptor. Temporal trajectory is the SIFT keypoint trajectory made by tracing the SIFT features of every representative frame, and describes the duration of visual content in the temporal domain, as well as the appearing or disappearing position. It is composed of a series of similar SIFT features arranged in chronological sequence. The descriptor of TCSIFT is the average of all SIFT descriptors in the temporal trajectory, i.e., a sort of visual content condensed for a period of time. Generally, as the video feature for retrieval, the descriptor of TCSIFT is used to retrieve. Thus, TCSIFT not only condenses visual content in temporal domain and reduces redundancy as well as the quantity of features, but also keeps the robustness and discrimination and records the video temporal characteristic.

TCSIFT is based on tracing the SIFT descriptors of representative frames. The tracing process of SIFT descriptors is the matching process of SIFT features. If the ratio of the distance between a descriptor and the second nearest neighbor to the distance between it and the first nearest neighbor is greater than the specific threshold, the SIFT descriptor matches the descriptor of its nearest neighbor:

$$\frac{dist(d, d_2)}{dist(d, d_1)} \geq \delta$$

$$dist(d_1, d_2) = \|d_1 - d_2\|_2, d \in D_{set_1}, d_1, d_2 \in D_{set_2}, d_1 \neq d_2 \quad (2)$$

where d_1 and d_2 are the SIFT descriptors of the first nearest neighbor and the second nearest neighbor of d in D_{set_2} , respectively. $dist(d, d_2)$ is the L2 (Euclidean) norm between d and d_2 , δ is a threshold, and D_{set_1}, D_{set_2} are the descriptor sets of frame₁ and frame₂ respectively. When tracing the SIFT descriptors, owing to the visual content changing over time, noise or camera motions, some SIFT descriptors disappear for a few frames during the process of tracing and appear again. As a result, a trajectory will be cut into a few segments. To solve this problem, these points that disappear temporarily and reappear need to be merged. However, the temporal trajectory should not last for many frames because the matching errors between two adjacent representative frames will propagate along the trajectory and increase. In this paper, the trajectory cannot exceed a certain number of frames. If the trajectory length is greater than a certain threshold, the trajectory will be split, until all trajectories are less than or equal to the threshold. The TCSIFT extraction consists of three steps:

Step1. Extract all SIFT of every representative frame. As is known, a SIFT keypoint is described by a geometric frame of four parameters: the keypoint center coordinates x and y , its scale (the radius of the region), and its orientation (an angle expressed in radians), as well as 128 dimension descriptors. We only use the 128 dimension descriptors and ignore the four parameters.

Step2. Trace every SIFT descriptor from the first representative frame using Algorithm 1. Then, the trajectory can be generated, as shown in Figure 2.

Fig.2. Schematic diagram of track

Algorithm 1. TCSIFT Extraction

Input: the SIFT features of every representative frame, the feature disappearing threshold θ_1 , and the length of a trajectory threshold θ_2 .

Output: TCSIFT

1. Generate trajectories. Track the SIFT descriptors frame by frame to generate temporal trajectories. A trajectory is denoted as $Tra_i = \{d_{fs}, \dots, d_{fi}, \dots, d_{fe}\}$, where f_i represents the representative frames where the tracked SIFTs lie and d_{fi} denotes the SIFT descriptor.
 2. Merge trajectories. For any two trajectories $Tra_i = \{d_{fs}, \dots, d_{fi}, \dots, d_{fe}\}$ and $Tra_j = \{d_{fb}, \dots, d_{fj}, \dots, d_{fo}\}$, if Tra_i matches with Tra_j and $t = f_b - f_s < \theta_1$, then merge the two trajectories as $Tra = \{d_{fs}, \dots, d_{fe}, d_{fb}, \dots, d_{fo}\}$, where θ_1 is a threshold.
 3. Split trajectories. For any $Tra_i = \{d_{fs}, \dots, d_{fi}, \dots, d_{fe}\}$, if the length of this trajectory is bigger than θ_2 , then $len(Tra_i) = n > \theta_2$. In this trajectory, the matching distance between the adjacent SIFT descriptors is denoted as $Dis_i = \{d_1, \dots, d_i, \dots, d_{n-1}\}$. If $\max(Dis_i) = d_i$, then split this trajectory into two trajectories from the middle of f_i and f_{i+1} .
-

4. Repeat step 2 and step 3 until all $len(Tra_i) < \theta_2$, where θ_2 is a threshold and as 15.

Step3. Calculate the descriptor of TCSIFT which is defined as the average of all SIFT descriptors in every temporal trajectory, according to the following formula:

$$TCSIFT = \hat{d} = \frac{1}{n} \sum_{f=f_s}^{f_e} d_f \quad (3)$$

where, f_s and f_e are the first and last frame of the trajectory, respectively, and n is the number of frames in the trajectory; d_f denotes the SIFT descriptor of the f_{th} representative frame appearing in the trajectory in the temporal domain. Some auxiliary information will be recorded such as the indexes of all of the descriptors and their frame IDs within the related trajectory.

Videos include temporal and spatial information. TCSIFT combines the similar SIFT features in the temporal domain, which not only reduces the feature redundancy to decrease the feature quantity but also contains the temporal information, which SIFT features themselves do not reflect. The TCSIFT proposed in this paper is based on tracking the SIFT features and is represented by the mean value of all SIFTs on the trajectory; thus, it retains basically the robustness and the discrimination of SIFT features and also includes most of the distinct features and compresses the redundant visual content to further reduce greatly the number of SIFT features. TCSIFT will be the basic retrieval cell. That is, given a TCSIFT, similar TCSIFTs in the feature database can be found.

4. Retrieval and Validation

In the retrieval process, TCSIFT is considered as the basic retrieval unit, and the whole retrieval process can be divided into two parts: similar TCSIFT retrieval and similar frame retrieval. Firstly, TCSIFT is extracted from the query clip using the TCSIFT extraction method as mentioned above. Secondly, a set of candidate similar TCSIFTs is retrieved from the feature database by using LSH. Thirdly, a set of candidate similar frames can be derived from the similar TCSIFT set for each query frame; some validations are done in this step. Here, we detail the implementation of our framework.

4.1. Similar TCSIFT retrieval

TCSIFT is regarded as the retrieval unit, and similar TCSIFT retrieval has two steps:

Step1. Determine search the scope. Given a query TCSIFT, the first step is to determine its searching scope by calculating the distances from the query to the K visual words. This is because the K-means is a hard clustering algorithm, which cannot ensure that the division of clusters is absolutely right. If the distances from a query TCSIFT to several visual words are almost the same, it is hard to determine which cluster the query TCSIFT should be assigned to. To ensure that the actual similar TCSIFTs are in the searching scope, we adopt the multi-assignment strategy as [36], in which the query TCSIFT is assigned to the previous three nearest visual words; as a result, the search scope is narrowed down to these three clusters..

Step2. Search for similar TCSIFT by hash mapping. In each of these three clusters, the K-NN searching will be done. In a cluster, the corresponding LSH functions and hash tables are loaded to memory and the hash key of the query TCSIFT is computed. The K-NN TCSIFT in the bucket with the same hash key will be returned as the candidate similar TCSIFT if its L2 distances to the query TCSIFT are not more than a certain threshold (which is set to 3000).

4.2. Matched frames mapping

After similar TCSIFT retrieval, every query TCSIFT gets a set of candidate similar TCSIFTs from the feature database. However, a frame is too small a unit to measure the similarity between two videos. As we know, every TCSIFT has a sequence of SIFT descriptors. We would like to do the SIFT matching between a query TCSIFT and a candidate TCSIFT, as shown in Figure 3(a). Assume that a query TCSIFT has the sequence of SIFT descriptors $D = \{d_{q1}, d_{q2}, \dots, d_{qn}\}$ and the candidate TCSIFT has the sequence of SIFT descriptors:

$\{d_{r1}, d_{r2}, \dots, d_{rn}\}$. A descriptor d_q is matched to the nearest descriptor d_r if:

$$\text{dist}(d_q, d_r) < \text{dist}(d_q, x) \quad x \in D, x \neq d_r \quad (4)$$

This matching is based on the prerequisites that the candidate TCSIFT is similar to the query TCSIFT and that all SIFT descriptors in one TCSIFT are similar based on the tracking algorithm. Thus, it is appropriate to match the d_q with d_r using the min distance.

After the SIFT descriptor matching, we group these SIFT descriptors in a query clip and reference video respectively according to their frame id to obtain the matched frames (see Figure 3(b)). For example, in Figure 3, the candidate matched frame of f_{q1} is f_{r1} ; the candidate matched frames of f_{q3} are f_{r2} and f_{r3} ; the candidate matched frames of f_{q4} are f_{r4} and f_{r5} .

4.3. Matched frames validation

After the similar frames retrieval step, each query frame has a set of candidate matching frames, but they may have false positives; thus, further validation is needed.

When an image rotates or its scale changes, the SIFT feature points extracted from this image also rotate by the same angle or have the same change in scale. Thus, the scale and orientation of SIFT features are consistent with the transformations of frames, which can be regarded as a measure for validating matched frames. Assume that the scale and orientation of SIFT feature point x , are $S(x)$ and $O(x)$, respectively; they will meet the following equations:

Fig.3. Obtained matched frames

$$\log(S(x_i^q)) - \log(s(x_i^r)) = C_1 \quad (5)$$

$$O(x_i^q) - O(x_i^r) = C_2 \quad (6)$$

where x_i^q is the i -th matched SIFT feature point from the query frame, x_i^r is the corresponding matched sift feature point from the candidate frame, and C_1 , C_2 are constants. In light of the scale space construction in the SIFT algorithm, a scale times a factor generates the next scale; thus, we use $\log(scale)$ instead of the scale itself to simplify the computation.

The validation processing algorithm is implemented as follows.

Algorithm 2. Validation

Input: the pair of matched frames

Output: the good pair of matched frames, from which most of the mismatches are removed.

1. Quantify $\log(S(x_i^q)) - \log(s(x_i^r))$ to 3 bins and $O(x_i^q) - O(x_i^r)$ to 8 bins.
2. Filter the matched SIFT pairs; only the pairs belonging to the bin with max number of pairs will remain.
3. Filter the pairs surviving from the second step. Only the pairs belonging to the 3 bins with max number of pairs will remain.
4. Keep the matched frames pair whose number of matched points is over 8.

After the validation by scale and orientation, most of the mismatched pairs will be removed, as shown in Figure 4. The pictures in the left column show the SIFT matching case before validation, and the right column shows that after validation, revealing that validation with the scale and orientation information of SIFT can remove most of the mismatches.

Fig.4. Validation of SIFT descriptors matching in matched frames

4.4. Re-rank the returned videos

The returned videos are represented by a sequence of frames, as shown in Figure 5. The numbers of matched frames in both the query clip and returned video should be more than a threshold (which is set as 5); otherwise, the video will not be returned. Each returned video gets a score to measure its matched degree to the query. The returned videos presented to users are ranked with respect to these scores.

Fig.5. Returned videos

The score calculation will consider two factors, namely content similarity and temporal consistency, and the score is the weighted sum of these two factors (see equation (7)). Content similarity is measured by using the average distance among all of the SIFT descriptors between the query clip and a returned video. If a sub-clip in returned video is similar to a sub-clip in the query clip, the timing order between the two sequences of frames in the two sub-clips should be consistent. For two matched sub-clips, $Clip_q = \{f_{q1}, f_{q2}, \dots, f_{qn}\}$ and $Clip_r = \{f_{r1}, f_{r2}, \dots, f_{rm}\}$, and the corresponding timestamps of every frame for each sub-clip are $TS_q = \{ts_{q1}, ts_{q2}, \dots, ts_{qn}\}$ and $TS_r = \{ts_{r1}, ts_{r2}, \dots, ts_{rm}\}$. The variance of the difference between TS_r and TS_q is used to measure the temporal consistency.

The returned videos presented to users are ranked in accordance with the similarity to the query. A score is calculated for each returned video to determine the ranking. The score of the returned videos is not only related to

the similarity between database videos and the query video but also the temporal consistency. If a database video contains or is similar to the query video clip, then the matched frames in this video should not be too dispersed, and the mapping between them and the query frames should be in consistent in time. The final score of the returned videos can be calculated as followings:

$$score = (2 \cdot sim) / 3 + 1 / (3 \cdot var) \quad (7)$$

where sim is the similarity of the reference video and query video just as mentioned above, and var is the variance of $\{timestamp_i^q - timestamp_i^r \mid i = 1, 2, \dots, n\}$. $timestamp_i^q$ and $timestamp_i^r$ are the timestamps of a query matched frame and matched frame in the database respectively. Before we start to calculate scores, sim and var should be normalized to 0~1. Here the weights (1/3 and 2/3) for sim and var are empirical values obtained by using mass tests.

5. Experimental Setup and Results

In this section, we conduct some experiments to show the effectiveness of our framework and algorithm. We first introduce the experimental datasets and then show and discuss our experimental results.

5.1. Dataset

We collect videos from two sources. The first source is the CC_WEB_VIDEO dataset [37] which is a Copy Web Video Dataset made by the VIREO group from City University of Hong Kong and the Informedia group from Carnegie Mellon University. In these datasets, 24 queries are selected to design to retrieve the most viewed and top favorite videos from YouTube and contain 12790 videos in total, approximately 732 hours and 90GB. The second source is the TRECVID 2014 IACC.1. The dataset [38] contains approximately 8000 Internet Archive videos (160 GB, 600 h) with durations between 10 s and 3.5 min. We comply the videos of the two sources into two datasets:

Dataset_A: 7700 representative videos are selected from CC_WEB_VIDEO and TRECVID to generate the test reference video database. These videos last from 1 minute to 6 minutes, totaling approximately 600 h, some of which are derived from the 24 query seed videos through some transformations.

Dataset_B: There are 5 classes of videos in this dataset: Camcording, Picture_in_picture, Photometric, Editing and Formatting which correspond to five video transformations. For each class, there are 24 query videos that can generate 120 video transformations in total. These videos last from 1 minute to 6 minutes.

Precision and recall are used to describe the performance of our system. In our video retrieval system, each returned video has a ranking that will be taken into consideration when evaluating. For example, given a query video clip, among the returned videos, we assume that the last ranked video which actually contains the query video clip, has the ranking m . The previous m returned videos are used to calculate the precision and recall.

$$precision = tp / rv, \quad recall = tp / pv \quad (8)$$

where tp is the number of true positive videos (i.e., containing the query clip) that are retrieved by our system, rv is the number of videos we retrieved from the dataset, and pv is the number of true positive videos in the dataset.

5.2. Experimental results

To validate whether the proposed methods are feasible and effective, four experiments have been conducted as follows. Firstly, experiment 1 will show the compression effect of TCSIFT compared to SIFT, and then experiment 2 and experiment 3 will give the performance comparison among the different methods. Finally, five types of video transformation are tested to evaluate the robustness of our framework to different video transformations.

1) High compressing property of TCSIFT

The TCSIFT proposed in this work combines similar SIFTs in the temporal domain, leading to an effective reduction of the quantity of features. In this experiment, five videos (from a few seconds to 6 minutes) are selected randomly to extract TCSIFT features and a comparison between the quantity of TCSIFT and SIFT is given to show the compression degree. In Figure 6, we note that the number of TCSIFT features was reduced by approximately 25% compared to that of SIFT; the compress ratio was much greater for slow changing videos, such as interviews.

Obviously, TCSIFT can directly reduce storage and computation cost, adapting easily to large-scale retrieval.

Figure 7 presents the results of the relation of δ to the compress ratio CR , which is represented by the following formula:

$$CR = n_{tcsift} / n_{sift} \times 100\% \quad (9)$$

where n_{sift} represents the number of SIFTs in a video and n_{tcsift} represents the number of TCSIFTs in a video. As δ , decreased, CR increased. While δ is equal to 1.2, the number of TCSIFT is half that of SIFT. However, δ that is too small can depress the discriminability of TCSIFT.

Fig.6. The number of SIFTs and TCSIFTs
#TCSIFT(1.5) and #TCSIFT(1.3) are the number of TCSIFTs when δ in formula (2) is set to 1.5 and 1.3, respectively

Fig.7. The different CR when δ is set to 1.2, 1.3, 1.4 or 1.5

2) Performance comparison

To validate the advantages of TCSIFT, the retrieval and speed performance comparison experiments are conducted here. In this experiment, Dataset_A is used as the test reference database. Three experiments based on three different feature extraction, i.e., TCSIFT, SIFT and visual-acoustic features (color and MFCC, abbreviated as VA features) [28], are done. In Figure 8 and Figure 9, three retrieval PR-curves and their retrieval times from 11 videos (from a few seconds to 6 minutes) are shown. In this experiment, considering stability and distinction, we adopt DOG detectors and SIFT descriptors. The implementation code of SIFT is downloaded from [39].

Fig.8. Comparison on precision and recall

Fig.9. Querying time comparison

As shown in Figure 8, it is obvious that the precision of TCSIFT ($\delta = 1.2$) is much higher than that of SIFT in most cases, but the retrieval recall of SIFT is better in some cases. This is because when matching features, if two TCSIFTs are similar, the SIFTs in these two TCSIFTs are always really similar. Furthermore, the redundant SIFT features, most of which are removed in TCSIFT, can disturb the matching process. However, the expression of average descriptor of TCSIFT may bring some errors that can cause mismatching between two similar TCSIFTs, and thus the recall will be slightly lower than that of SIFT. Considering both precision and recall, their performances are not far-off, but there exists a problem of trade-off between precision and recall. Figure 9 shows that the retrieval speed of TCSIFT is obviously better than that of SIFT, which is because the quantity of TCSIFTs is at least 25% less than that of SIFTs within the same video, which can help to accelerate the retrieval greatly.

According to [28], this proposed method jointly employs color-based visual features and audio fingerprints for detecting duplicate videos. From Figure 8, we can see that precision of TCSIFT is much greater than that of VA

features, but the recall is slightly less than that of VA features. The audio content is a significant information source of a video sequence and can provide some auxiliary clues for judging video copy detection; thus, the recall will be a little higher. However, to calculate the audio fingerprints and save them, it is unavoidable to afford extra time and space. On the other hand, the much smaller searching scope is exactly the reason why the retrieval speed of TCSIFT is almost 100 times faster than that of VA features, as shown in Figure 9.

Table 2 lists the mean memory size and average of matching time of TCSIFT and SIFT. When scanning over a video, TCSIFT skips more representative frames without computing the complex actual similarity. According to Equation (2), δ for TCSIFT is proportional to the difference between the actual similarity and the threshold. As illustrated in Figure 8, δ is set between 1.2 and 1.5; thus, the performance will be better.

Table 2. Comparison on matching time and memory consumption

	The average of matching Time (s)	Memory (MB)
SIFT	1.187	71.6
TCSIFT	0.678	34.7

As shown in Table 3, compared to the classic matching validation method RANSAC [40], the validation method, i.e., Algorithm 2, has much lower complexity and is fast. In some cases such as local transformation and strong distortion, this method may not achieve results as good as those in Figure 4, but these cases rarely appear, and our method works well in most cases. In Table 3, AVT_V and AVT_R denote the average validation time of our validation algorithm (Algorithm 2) and the average validation time of the RANSAC algorithm, respectively. MP_V denotes the number of mismatched pairs reduced by our validation algorithm, and MP_R denotes the number of mismatched pairs reduced by the RANSAC algorithm.

Table 3. Comparison on the average validation time and the number of mismatched pairs reduced

	AVT_V/AVT_R (%)	MP_V/MP_R (%)
Our validation algorithm vs RANSAC	81.53	87.27

3) Comparison among five kinds of video transformation

Figure 10 shows the retrieval recall and precision. All five types of video transformation achieved a retrieval precision over 0.65; the three transformations (Photometric, Formatting and Editing) achieved a recall over 0.7; Camcording achieved a low recall because of the weak robustness of SIFT to strong distortion in frames. When a strong Camcording occurs, the invariance of a part of the SIFT feature gets destroyed; thus these features are hard to find when retrieving. On the other hand, if features are found, it means that they are strong enough and distinctive; thus, they are always really matched ones. The recall of Picture_in_picture will rely on the portion of real content out of the whole frame to a certain extent, and the greater the portion, the better the performance. By the same token, the less interference, the better for Photometric transformation of video retrieval. Strong interference causes a low matching rate between similar features, so the retrieval precision can be low. Editing and Formatting transformations will not change feature attributes, and so both of these two transformations have relatively good precision and recall. However, in some cases, such as covering with a photo-frame or adding a logo to videos when

Fig.10. PR curves of five types of video transformations

doing an editing transformation, some features cannot be extracted because of the shelter, leading to the decline of precision and recall.

6. Conclusion

To address the efficiency problem of video copy retrieval in a large-scale repository of video files, we propose that the TCSIFT strategy helps to highly compact features to decrease the quantity of features but maintains most of the distinctive features by tracking SIFT and encoding the temporal information; thus, we design a framework based on TCSIFT and improve the retrieval and validation process at the feature and frame level. In view of these advantages, TCSIFT can also be suitable for video content protection, commercial detection, topic tracking, piracy removal, search result re-ranking, clustering, and so on. Experimental results demonstrate that for various video transformations, the proposed framework and algorithm are successfully integrated to yield comparable accuracy, compact memory consumption, and efficient execution time.

Acknowledgements

The authors wish to acknowledge the financial support from: (i) Special funds for the development of strategic emerging industries in Shenzhen (JCYJ20130326105637578) and Shenzhen science and technology plan project (JCYJ20150324141711630); and (ii) National Natural Science Foundation of China (61502311 and 61501299).

References

- [1] Datta R, Li J, Wang J Z., Content-based image retrieval: approaches and trends of the new age, in: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval. ACM, 2005, pp. 253-262.
- [2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, Content-based image retrieval at the end of the early years, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12) (2000) 1349-1380.
- [3] Richang Hong, Meng Wang, Yue Gao, Dacheng Tao, Xuelong Li, Xindong Wu, Image annotation by multiple-instance learning with discriminative feature mapping and selection. *IEEE T. Cybernetics* 44(5) (2014) 669-680.
- [4] Tan, Hung-Khoon, Chong-Wah Ngo, Richard Hong, and Tat-Seng Chua, Scalable detection of partial near-duplicate videos by visual-temporal consistency, in: *Proc. ACM Conf. Multimed.*, 2009, pp. 145-154.
- [5] Y. Meng, E. Y. Chang, B. Li. Enhancing dpf for near-replica image recognition, in: *Proc. of Int. Conf. on Pattern Recognition, ICPR 2003*, pp. 416-423.
- [6] Law-To J, Chen L, Joly A, et al., Video copy detection: a comparative study. in: *Proceedings of the 6th ACM international conference on Image and video retrieval. ACM, 2007*, pp. 371-378.
- [7] Ke Y, Sukthankar R, Huston L, et al., Efficient near-duplicate detection and sub-image retrieval, *ACM Multimedia*. 4(1) (2004) 5.
- [8] Liu, J., Huang, Z., Cai, H., Shen, H. T., Ngo, C. W., Wang, W. Near-duplicate video retrieval: Current research and future trends. *ACM Computing Surveys (CSUR)*, 45(4), (2013) 44.
- [9] Yuan J, Duan L Y, Tian Q, et al. Fast and robust short video clip search using an index structure, in: *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval. ACM, 2004*, pp. 61-68.
- [10] Richang Hong, Jianxin Pan, Shijie Hao, Meng Wang, Feng Xue, Xindong Wu. Image quality assessment based on matching pursuit. *Inf. Sci.* 273 (2014) 196-211.
- [11] Apostol Natsev, John R. Smith, Matthew Hill, Gang Hua, Bert Huangy, Michele Merlery, Lexing Xie, Hua Ouyangz, Mingyuan Zhou, IBM research TRECVID-2010 video copy detection and multimedia event detection system. in: *Proceedings of NIST TRECVID, Workshop, 2010*.
- [12] Jian Li, Xiaolong Li, Bin Yang, Xingming Sun, Segmentation-based Image Copy-move Forgery Detection Scheme, *IEEE Transactions on Information Forensics and Security*, 10(3)(2015) 507-518.
- [13] Weiming Hu, Dan Xie, Steve Maybank, Semantic-Based Surveillance Video Retrieval, *IEEE trans. on image processing*. 16(4)(2007) 1168-1181.
- [14] Tasdemir K, Cetin A E. Motion vector based features for content based video copy detection. in: *ICPR, 2010*, pp. 3134-3137.
- [15] Richang Hong, Zheng-Jun Zha, Yue Gao, Tat-Seng Chua, Xindong Wu, Multimedia encyclopedia construction by mining web knowledge.

Signal Processing 93(8) (2013) 2361-2368.

- [16] Huang, Z., Hu, B., Cheng, H., Shen, H. T., Liu, H., AND Zhou, X. Mining near-duplicate graph for cluster-based reranking of web video search results. *ACM Trans. Inf. Syst.* 28(4) (2010) 22.
- [17] Huang, Z., Shen, H. T., Shao, J., Zhou, X., and Cui, B. Bounded coordinate system indexing for real-time video clip search. *ACM Trans. Inf. Syst.* 27(3) (2009)17–33.
- [18] Wu, X., Takimoto, M., Satoh, S., and Adachi, J. Scene duplicate detection based on the pattern of discontinuities in feature point trajectories. In: in: *Proc. ACM Conf. Multimed*, 2008, pp.51–60.
- [19] Zhao, W., NGO, C.W. Scale-rotation invariant pattern entropy for keypoint-based near duplicate detection. *IEEE Trans. Image Process.* 18(2) (2009)412–423.
- [20] Jiang, Y.G., NGO, C.W. Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval. *Comput. Vis. Image Understand.* 113(3) (2009) 405–414.
- [21] Douze, M., Jegou, H., and Schmid, C. An image-based approach to video copy detection with spatiotemporal post-filtering. *IEEE Trans. Multimedia* 12(4) (2010)257–266.
- [22] B. Coskun, B. Sankur, N. Memon. Spatio-temporal transform based video hashing. *IEEE Trans. on Multimedia*, 8(6) (2006)1190–1208.
- [23] Mani Malekesmaeili, Mehrdad Fatourehchi, Rabab K. Ward. Video copy detection using temporally informative representative images. In: *International Conference on Machine Learning and Applications*,2009, pp.69-74.
- [24] Guo Junbo, Li Jin tao, Zhang Yongdong, et al .Video copy detect ion based on trajectory behavior pattern. *Journal of Computer Aided Design and Computer Graphics*, 22 (6) (2010)943 -948.
- [25] Wu Xiao, Li Jintao et al.Video copy detection based on spatio-temporal trajectory behavior feature. *Journal of Computer Research, Development.* 47(11) (2010)1871-1877.
- [26] Willems G, Tuytelaars T, Van Gool L. An efficient dense and scale-invariant spatio-temporal interest point detector.in:*Proc. ECCV 2008*. pp.650-663.
- [27] Law-to, J., Buisson, O., Gouet-brunet, V., and Boujemaa, N. Robust voting algorithm based on labels of behavior for video copy detection. in: *Proc. ACM Conf. Multimed*,2006, pp. 835–844.
- [28] R. Roopalakshmi. A novel framework for CBCD using integrated color and acoustic features. *International Journal of Multimedia Information Retrieval*. 1-13(2014).
- [29] Roopalakshmi R, Ram Mohana Reddy G. A novel spatio-temporal registration framework for video copy localization based on multimodal features. *Signal processing*, 93(8) (2013) 2339-2351.
- [30] L. Shang et al., Real-Time large scale near-duplicate video retrieval, in: *Proc. ACM Conf. Multimed*, 2010,pp. 531-540.
- [31] W.-L. Zhao, S. Tan, and C.-W. Ngo, Largescale near-duplicate web video search:challenge and opportunity,in: *Proc. IEEE Int’l Conf. Multimedia and Expo (ICME)*, IEEE, 2009,pp. 1624-1627.
- [32] Woogyoung Jun, Yillbyung Lee and Byoung-Min Jun, Duplicate video detection for large-scale multimedia, *Multimed Tools Appl* 06(2015)131-144.
- [33] Alexis Joly, Olivier Buisson and Carl Frelicot, Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Trans. on Multimedia*, 9(2) (2007)293-306.
- [34] AlexandrAndoni and Piotr Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, *Communications of the ACM*, 51(1) (2008)117-122.
- [35] A.Anjulana and N.Canagarajah. A unified framework for object retrieval and mining. *IEEE Trans. on Circuits and Systems for Video Technology*, 19(1) (2009)63-76.
- [36] Douze M, Jégou H, Schmid C. An image-based approach to video copy detection with spatio-temporal post-filtering. *Multimedia, IEEE Transactions on*, 12(4) (2010)257-266.
- [37] CC_WEB_VIDEO: Near-Duplicate Web Video Dataset: <http://vireo.cs.cityu.edu.hk/webvideo/>.
- [38] Guidelines for TRECVID 2014:<http://www-nlpir.nist.gov/projects/tv2014/tv2014.html>
- [39] VLFeat: <http://www.vlfeat.org>.
- [40] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) (1981) 381-395.

Figure 1.docx

Figure 3(a).eps

figure 4.eps

figure 5.eps

Figure 6.tif

figure 7.eps

figure 8.eps

figure 9.eps

figure 10.eps