An Approach to: Identify and Retrieve Subsequence Matching Frames from Videos

[1]R Shiva Shankar, [2]D Ravi Babu, [3]J Rajanikanth, [4]K V S Murthy
[1,4]Dept. of CSE, SRKR Engineering College, Bhimavaram, West Godavari, AP, India

Abstract
Retrieving a duplicate video is becoming more and more important with the exponential growth. Though various approaches have been proposed to manipulate a large video database, effective video indexing and retrieval, these are mainly focusing on the retrieval accuracy while infeasible to query on Web scale video database in real time. This paper proposes a novel method for efficiency and scalability issues for similar frames through video. We proposed a novel method to inserting a distinct pattern into the video stream, video copy detection techniques match content-based signatures to detect copies of video. Existing typical content-based copy detection schemes have relied on image matching. To effectively match the video sequences with a low computational load, we use the key frames extracted by the cumulative directed divergence and compare the set of key frames using the novel method and g-l technique. Motion, intensity and color-based signatures are compared in the context of copy detection. Results are reported on detecting copies of movie clips. To demonstrate the effectiveness and efficiency of the proposed method, we evaluate its performance on an open video data set containing about 10K videos and compare it with four existing methods in terms of precision and time complexity.

Keywords
Database, Video Retrieval, Sequence Matching, Video indexing

I. Introduction

A video sequence is an ordered set of a large number of frames, and from the database research perspective, each frame is usually represented by a high-dimensional vector, which has been extracted from some low-level content features, such as color distribution, texture pattern, or shape structure within the original media domain. Matching of videos is often translated into searches among these feature vectors [1-2].

Nowadays, the rapid advances in multimedia and network technologies popularize many applications of video databases, and sophisticated techniques for representing, matching, and indexing videos are in high demand. In practice, it is often undesirable to manually check whether a video is part of a long stream by browsing its entire length [3-4]; thus, a reliable solution of automatically finding similar content is imperative.

Video subsequence identification involves locating the position of the most similar part with respect to a user-specified query clip Q from a long pre-stored video sequence S. Ideally; it can identify relevant video, even if there exist some transformation distortion, partial content reordering, insertion, deletion, or replacement [5,6]. Its typical applications include:

- Recognition of Copyright enforcement: Owners of video content would like to be aware of any use of their data, in any media or representation.
- Detecting TV Commercial: Some companies would like to track their TV commercials when they are aired on different channels during a certain time period for statistic purpose.

The primary difference between video retrieval and video subsequence identification is that, while retrieval task conventionally returns similar clips from a large collection of videos which have been either chopped up into similar lengths or cut at content boundaries, subsequence identification task aims at finding if there exists any subsequence of a long database video that shares similar content to a query clip.

The studies on content-based copy detection are concerned with a specific problem of co-derivative video detecting a graph transformation and matching approach to process variable length comparison over database video with query [7-9]. It facilitates safely pruning a large portion of irrelevant parts and rapidly locating some promising candidates for further similarity evaluations. Constructing a bipartite graph representing the similar frame mapping relationship between Q and S with an efficient batch kNN search algorithm (k Nearest Neighbors), video subsequences can be extracted.

Our robust model for evaluating video similarity is based on the percentage of similar frames as well as measuring the problem of video similarity. Finally our method can better resemble the "similar" notion of human perception.

II. Background

Video near-duplicate retrieval has become a compelling researching topic in recent years, due to the proliferation of online video uploading/sharing sites and the exponential explosion of video data. To full fill this requirement several authors proposed various techniques. In this section we discuss some of them [10].

Yuan et al. [11] provided a list of all query-by-video-clip methods developed in the past decade. According to different user requirements, video search intentions can be generally classified into three main categories: detecting instances of copies, identifying visual similarities of near duplicates, and retrieving entries in a broad sense of high-level concept similarity. The following discussions focus on the first and second categories. The third category that exploits semantic meaning is beyond the scope of this work.

A. Detection of Duplicate Video

Extensive research efforts have been made on extracting and matching content-based signatures to detect copies of videos. Mohan [12] introduced to employ ordinal measure for video sequence matching. Naphade et al. [5] developed an efficient scheme to match video clips using color histogram intersection. Pua et al. [6] proposed a method based on color moment feature to search video copy from a long segmented sequence. Hampapur et al. [13] examined several methods of using a sequence of frame features to leverage the characteristic of sequence-to-sequence matching. In their work, query sequence slides frame by frame on database video with a fixed length window. In addition to distortions introduced by different encoding parameters, Kim and Vasudev [2] proposed to use spatiotemporal ordinal signatures of frames to further address display format conversions, such as different aspect ratios.

Since the process of video transformation could give rise to several distortions, techniques circumventing these variations by globe signatures have been considered. They tend to depict a

video globally rather than focusing on its sequential details. Some properties that are likely to be preserved even with these variations (e.g., shot length information) were suggested to be generated as compact signatures and string matching technique could be used to report such a copy [14]. This method is efficient, but has limitations with blurry shot boundaries or very limited number of shots. Moreover, in reality, a query video clip can be just a shot or even a sub shot. However, this method is only applicable for queries which consist of multiple shots.

B. Video Sequence Similarity Search

The video sequence similarity matching problem can be stated as follows: given two video sequences, determine how similar the two sequences are. Formally, given two video sequences: V1, V2, and some possible variation in the frames making up the sequences, we wish to find some function that will take V1 and V2 as its input, and return some indicator of their similarity. Then we can compare the similarity indicator with a pre-stated threshold of similarity, and declare V1 and V2 similar if they pass the threshold. Unlike in the case of exact matching, since we are interested only in similarity between the sequences, the V1 and V2 need not be of the same length, and the size of the frames in each sequence need not be the same. Conceptually, V1 can represent the entire database, while V2 is just a short query sequence. Here, we would be interested in knowing if there exists any subsequence of V1 that is similar to V2. In video
Sequence matching, three basic types of matching can be identified:

- scene-to-scene: check if two scenes are similar;
- scene-to-sequence: check if a scene similar to the query scene occurs in the database sequence;
- sequence-to-sequence: check if a sequence similar to a query sequence occurs in the database sequence. The query can contain more than one scene.

To handle the trivial case of sequence-to scene matching, we simply assume that the database sequence is the longer of the two sequences, though this has no effect on the actual matching process. We observe that (ii) is a generalization of (i), and will make use of methods for (i). Similarly, (iii) is a generalization of (ii) and its solution will depend on solutions to (ii). The basic problem thus is finding solutions to (i): the scene-to-scene matching problem..

III. System Architecture

Despite the importance, video subsequence identification, which is to find the similar content to a short query clip from a long video sequence, has not been well addressed. This paper presents a graph transformation and matching approach to this problem, with extension to identify the occurrence of potentially different ordering or length due to content editing. With a novel batch query algorithm to retrieve similar frames, the mapping relationship between the query and database video is first represented by a bipartite graph. The densely matched parts along the long sequence are then extracted, followed by a filter-and-refine search strategy to prune some irrelevant subsequences. In this paper the input video is nothing but a small video clip which is to find the similar contents from a database video. When an input video is tacked then it will be split into frames by using SIF algorithm. The mapping relationship between the query and database video is first represented by a bipartite graph.

The intent and purpose of this paper is to present the requirement specifications for the software product entitled "Video Subsequence Identification". We show vulnerabilities of video copies and video similarities and propose an effective and efficient way of query processing for video subsequence identification. Fig 1 shows our proposed system architecture.

The main idea behind the identification of the subsequence frames is the comparison of the query frames with that of the database frames. The main processing starts with splitting of the videos to frames. The frames are mapped using a mapping relationship based on batch query algorithm and the most densely mapped frames can be obtained by constructing the bipartite graph discussing in later sections.

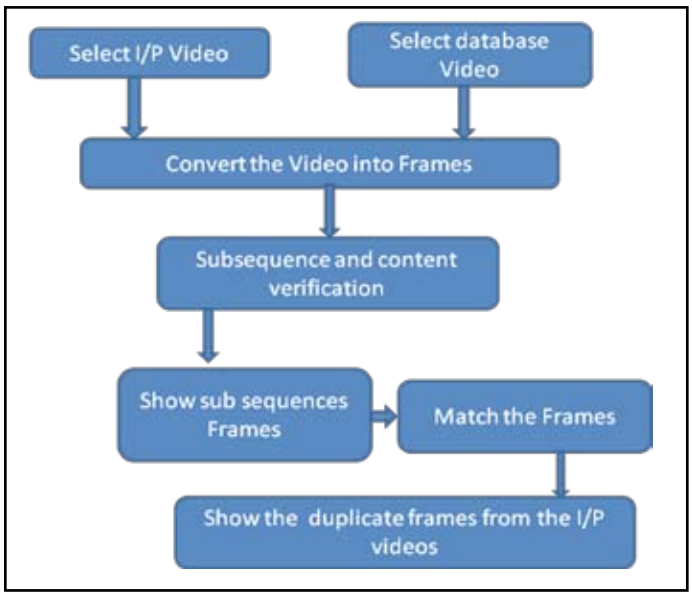
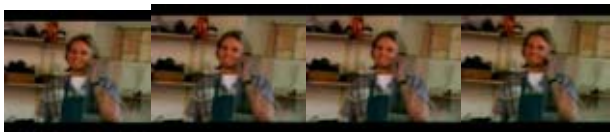


Fig. 1: System Architecture

A. Retrieve Similar Frames

In conventional content-based similarity search, a query consumes a single NN search by traversing the indexing structure once. However, a distinguishing characteristic of video retrieval is that, each video is described by a sequence of feature vectors, so as to the query. A query clip and a database video are taken and feature vectors are checked to identify whether the database frame is similar to query frame or contains query frame, typically for each, a search is first performed to retrieve the similar feature vectors. A typical video similarity measure is to compute the percentage of similar feature vectors shared by two videos and, their similarity is defined. Retrieval of similar feature vectors is processed as a range or search in high-dimensional space.





B. Proposed Algorithm

A Batch Query Algorithm is used for the retrieval of similar frames. It takes a query frame and the database video as input. A kNN (k-Nearest-Neighbour) search is employed on the database frames and the query frame and hence, all the database frames that match with the query frame are saved in a set. For all the other frames of the database video that do not obey kNN search but are in the

range of the query frame, they are also added to the set of similar frames. Hence, this algorithm provides exactly matched frames as well as similar frames of the database video with that of the query frame.

Step 1: Select input video
Step 2: Select database video
Step 3: Convert input video into number of frames
Step 4: Convert database video into number of frames
Step 5: Sub sequence identification
Step 6: Match the frames
Step 7: Shows the duplicate frames from input videos

To retrieve similar frames we have follow procedures as shown in Table 1.

Table 1: Psuedo Code for Retrieve Similar Frames

```
Begin
    Step 1: consider 3 vectors of String type.
    Step 2: Initialize the vectors.
    Step 3: Take Substrings of strings from vectors/
    Step 4: Append them by"==" and add it it third
            vector.
    Step 5: Repeat Step 3 & Step 4.
    Step 6: Pass the 3rd Vector to set ListData( ).
end
```

1. Discussion

Here, a video is first divided into a number of elementary segments, and the feature of each segment is then extracted. These features of the videos can have both spatial and temporal features. Spatial features include still image features, such as color, shape, texture, etc. Temporal features consist of object motion, camera operation, etc. Normally, all the features are represented by high-dimensional vectors. In short, each video is translated to a sequence of high-dimensional feature vectors. For a large video database, scanning on all these vectors is strongly undesirable due to the high complexity of video features. The complexity can be reduced to a level that can be efficiently managed. The compact video representations are organized in a way such that fast retrieval is feasible.

C. Bipartite Graph Transformation

For most general videos, retrieval based on a single shot may not be practical, since a shot itself is only a part of an large video. The context can be an event, activity, or a physical location. Basically, shot-based retrieval is useful for tasks such as the detection of known objects and certain kinds of videos like sports. For most casual users, query-by-clip is definitely more concise and convenient than query-by-shot.

The similarity ranking of clips, in general, is built upon the shot-based retrieval. Besides relying on shot similarity, clip similarity is also dependent on the interrelationship such as temporal order, granularity, and interference among shots. Granularity models the degree of one-to-one shot matching between two clips, while interference models the percentage of unmatched shots.

We adopt bipartite graph matching algorithms matching of shots in clips. The algorithm is constrained under one-to-one mapping. By computing the maximum cardinality of matching, it is capable of rapidly filtering irrelevant clips. The goal of clip filtering is to rapidly prune irrelevant clips and thus improve the speed efficiency of retrieval. By optimizing the total weight of matching, it is able to rank relevant clips based on the similarity of visual and granularity. Thus it forms a hierarchical framework for filtering and retrieval. To construct Bipartite Graph we have to follow procedure shown in Table 2.

Table 2: Pseudo code for Construction of Bipartite Graph

Begin
Step 1: Repeat Step 2,3,4 & 5 until the last frame of video
Step 2: Read the current values of file
Step 3: Add current value to current mean.
Step 4: Increment Current count.
Step 5: Append current character to current image.
Step 6: Calculate current mean.
Step 7: Display mean and count of both images.
Step 8: Set boundaries and display
Step 9: Set distance based on precount and boundaries
Step 10: Display distance
end;

1. Discussion

The output of the batch query algorithm becomes the input for this algorithm. While similar frame retrieval is the preliminary step analogous to building element-to-element correspondence, this algorithm emphasizes on effective temporal localization of target subsequence. Each frame of the query video and the database video can be placed as a node along a 1D-temporal line and are treated as two sides of a bipartite graph. Matching is done between the two sets of nodes and the densely matched frames are saved as similar frames and the remaining sparsely matched and unmatched frames are discarded.

IV. Results

We make use of the videos of MPEG format and an average frame rate of 25fps is obtained. As a starting point, the most prominent visual feature RGB color values were extracted due to its computation efficiency and provide useful clues for similarity of frames. Since video content evolution within a second is usually negligible, for simplicity, from the original sequence of video frames, the first one of every 25 frames is further extracted, i.e., on average one frame is sub sampled from each second of the database video. This choice is efficient for fast search, but also fine enough to accurately locate a target subsequence. Each query is a short video (varying from 5 to 10 seconds) taken from the test video recording, optionally edited with some change to make it have different ordering, length, or content with the original extracted fragment (i.e., ground truth). Every time a query video is submitted to identify the starting and ending boundaries of its corresponding target subsequence in the database.

The input and the database videos are selected and then players are started. The video format, frame dimensions, frame rate and the length of the video are identified. The videos are split to frames at a frame rate of 25 fps and hence, the frame no., the time at which the frame is identified and the length of the frame is displayed. The mean measures of the frames are calculated and the frames which come within a range of min and max values are identified as similar frames and their distance is taken as 0. The mean values of each pair of similar frames can also be known. Here we show the proposed interface.

Fig. 2 shows the Subsequence Identification window. It shows the splitted frames are matched with one another and the similar frames that are identified are shown as subsequence detected frames. Fig 3 shows the subsequence identified frames window, In this, the similar frames that are identified from the query and the database videos are displayed as a list.



Fig. 2: Subsequence Identification

Fig . 4 shows viewing the similar frames window, In this, the similar frames that are displayed in the list can be viewed by clicking on a pair of identified frames.

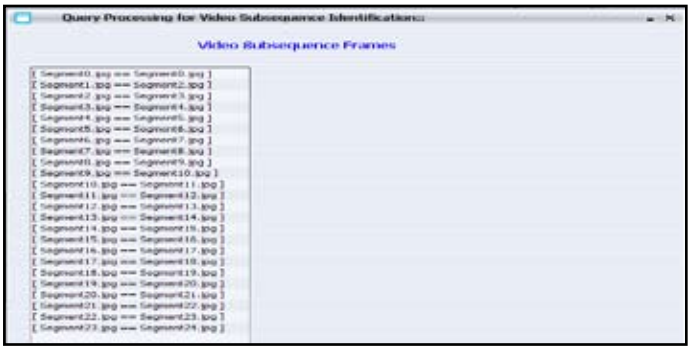


Fig. 3: Subsequence Identified Frames

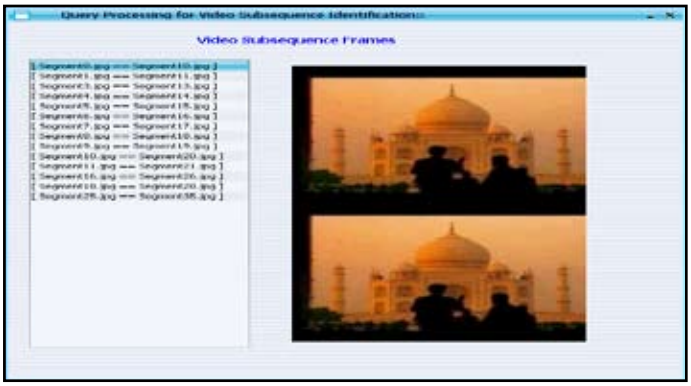


Fig. 4: Viewing the Similar Frames

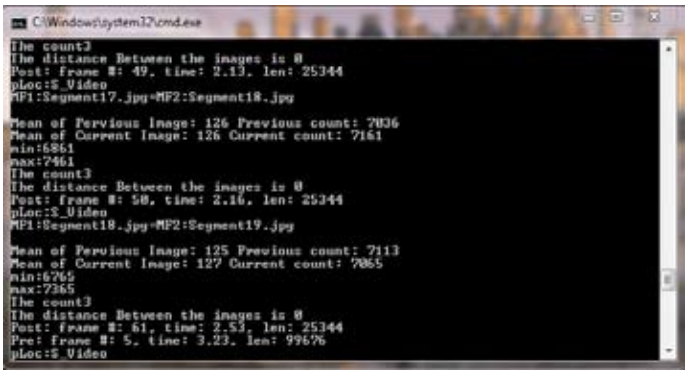


Fig. 5: Analysis of the Displayed Similar Frames

V. Conclusion

Now a days, demand on visual information of rich content, effective and efficient manipulations of large video databases are increasingly desired. Many investigations have been made on content-based video retrieval. However, despite the importance, video subsequence identification, which is to find the similar content to a short query clip from a long video sequence, has not been well addressed. This paper presents a video sequence identification approach that can reliably and quickly detect equal or similar recurrences of a given video sequence in long video streams, e.g. such as TV broadcasts.

References

[1] X. Liu, Y. Zhuang, Y. Pan,"A New Approach to Retrieve Video by Example Video Clip", Proc. Seventh ACM Int'l Conf. Multimedia (MULTIMEDIA '99), Vol. 2, pp. 41-44, 1999.

[2] Y. Wu, Y. Zhuang, Y. Pan,"Content-Based Video Similarity Model", Proc. Eighth ACM Int'l Conf. Multimedia (MULTIMEDIA '00), pp. 465-467, 2000.

[3] R. Mohan, Video sequence matching", in Proc. Intl. Conference on Speech, Accoustics and Signal Processing,1998.

[4] A. K. Jain, A. Vailaya, W. Xiong, Query by video clip", Multimedia Systems, 1999.

[5] M. M. Yeung, B. Liu, Efficient matching and clustering of video shots", in Proceedings of the IEEE Intl.Conference on Image Processing, Vol. 1, pp. 338-341, (Washington, D.C.), Oct. 1995.

[6] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain,"Content-Based Image Retrieval at the End of the Early Years", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, pp. 1349-1380, Dec. 2000.

[7] C. Kim, B. Vasudev,"Spatiotemporal Sequence Matching for Efficient Video Copy Detection", IEEE Trans. Circuits and Systems for Video Technology, Vol. 15, No. 1, pp. 127-132, 2005.

[8] X.-S. Hua, X. Chen, H. Zhang,"Robust Video Signature Based on Ordinal Measure", Proc. IEEE Int'l Conf. Image Processing (ICIP '04), pp. 685-688, 2004.

[9] C.-Y. Chiu, C.-H. Li, H.-A. Wang, C.-S. Chen, L.-F. Chien, "A Time Warping Based Approach for Video Copy Detection", Proc. 18th Int'l Conf. Pattern Recognition (ICPR '06), vol. 3, pp. 228-231, 2006.

[10] M.R. Naphade, M.M. Yeung, B.-L. Yeo,"A Novel Scheme for Fast and Efficient Video Sequence Matching Using Compact Signatures", Proc. Storage and Retrieval for Image and Video Databases (SPIE '00), pp. 564-572, 2000.

[11] K.M. Pua, J.M. Gauch, S. Gauch, J.Z. Miadowicz,"Real Time Repeated Video Sequence Identification", Computer Vision and Image Understanding, Vol. 93, No. 3, pp. 310-327, 2004.

[12] K. Kashino, T. Kurozumi, H. Murase,"A Quick Search Method for Audio and Video Signals Based on Histogram Pruning", IEEE Trans. Multimedia, Vol. 5, No. 3, pp. 348-357, 2003.

[13] S.C.S. Cheung, A. Zakhor,"Efficient Video Similarity Measurement with Video Signature", IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, No. 1, pp. 59-74, 2003.

[14] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, H. Sun,"Survey of Compressed-Domain Features Used in Audio-Visual Indexing and Analysis", J. Visual Comm. and Image

Representation, Vol. 14, No. 2, pp. 150-183, 2003.
[15] H.T. Shen, B.C. Ooi, X. Zhou, Z. Huang,"Towards Effective Indexing for Very Large Video Sequence Database", Proc. ACM SIGMOD '05, pp. 730-741, 2005
[16] J. Yuan, L.-Y. Duan, Q. Tian, S. Ranganath, C. Xu,"Fast and Robust Short Video Clip Search for Copy Detection", Proc. Fifth IEEE Pacific-Rim Conf. Multimedia (PCM '04), Vol. 2, pp. 479-488, 2004.
[17] A. Hampapur, K.-H. Hyun, R.M. Bolle,"Comparison of Sequence Matching Techniques for Video Copy Detection", Proc. Storage and Retrieval for Image and Video Databases (SPIE '02), pp. 194-201, 2002.
[18] E.J. Keogh,"Exact Indexing of Dynamic Time Warping", Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), pp. 406-417, 2002.
[19] M. Vlachos, D. Gunopulos, G. Kollios,"Discovering Similar Multidimensional Trajectories", Proc. 18th Int'l Conf. Data Eng. (ICDE '02), pp. 673-684, 2002.
[20] Y. Peng, C.-W. Ngo,"Clip-Based Similarity Measure for Query-Dependent Clip Retrieval and Video Summarization", IEEE Trans. Circuits and Systems for Video Technology, Vol. 16, No. 5, pp. 612-627, 2006.