Video Copy Detection by Fast Sequence Matching

Mei-Chen Yeh
Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106, USA
1-805-893-4852
meichen@umail.ucsb.edu

Kwang-Ting Cheng
Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106, USA
1-805-893-4852
timcheng@ece.ucsb.edu

ABSTRACT
Sequence matching techniques are effective for comparing two videos. However, existing approaches suffer from demanding computational costs and thus are not scalable for large-scale applications. In this paper we view video copy detection as a local alignment problem between two frame sequences and propose a two-level filtration approach which achieves significant acceleration to the matching process. First, we propose to use an adaptive vocabulary tree to index all frame descriptors extracted from the video database. In this step, each video is treated as a "bag of frames." Such an indexing structure not only provides a rich vocabulary for representing videos, but also enables efficient computation of a pyramid matching kernel between videos. This vocabulary tree filters those videos that are dissimilar to the query based on their histogram pyramid representations. Second, we propose a fast edit-distance-based sequence matching method that avoids unnecessary comparisons between dissimilar frame pairs. This step reduces the quadratic runtime to a linear time with respect to the lengths of the sequences under comparison. Experiments on the MUSCLE VCD benchmark demonstrate that our approach is effective and efficient. It is 18X faster than the original sequence matching algorithms. This technique can be applied to several other visual retrieval tasks including shape retrieval. We demonstrate that the proposed method can also achieve a significant speedup for the shape retrieval task on the MPEG-7 shape dataset.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, search process.* I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision – *Applications.*

General Terms
Algorithms.

CIVR'09, July 8-10, 2009 Santorini, GR

Keywords
Video copy detection, local alignment, similarity measure, vocabulary tree.

1. INTRODUCTION
With digital video content production and distribution continuing to grow, content-based copy detection (CBCD) has been actively studied for a wide range of applications that include searching [8, 20], multimedia linking [4, 23], and protecting copyrighted content [5, 6, 7, 10, 11, 15, 16]. Based on content alone, CBCD attempts to identify segments in a query video that are *copies* from a reference video database. A copy is not an exact duplicate but, in general, either a transformed or a modified version of the original document that remains recognizable [6]. Transformations to digital content such as cropping and inserting logos are frequently performed and the resulting near-duplicates could be different from the source in terms of not only formats, but also content [20].

Many existing approaches cast the task of CBCD into a traditional content-based key-frame retrieval framework [5, 11] since both tasks follow the query-by-example paradigm [6]. However, CBCD aims at identifying video copies instead of similar individual frames. For example, two videos of the same scene may be considered similar; however, they are not necessarily *copies* of each other (based on the definition described above). Thus, methods that solely rely on frame-level similarities can suffer from high false positive rates [7].

Since a video can be naturally represented as a sequence of frames, temporal constraints have been employed in the design of metrics that compare the similarities between two videos [1, 3, 8, 24]. More specifically, videos are represented as strings of symbols and the *edit distance* between two symbol strings—defined as the minimal cost of any insertions, deletions, and substitutions of symbols needed to make two strings equal—is used for measuring video similarity. Video matching methods based on such a metric have a number of merits. First, the *ground distance* used to compare frame descriptors can be seamlessly integrated into the distance measurement. Second, the temporal order is preserved during matching. Moreover, two similar videos that differ either in length, or in terms of other factors such as differences in subsequences caused by incorrect key frame detection, are likely to obtain a high similarity score based on

such a metric. Recent studies have shown some successes in the use of the edit distance in the context of video clip retrieval [3].

However, two major problems hinder the use of the edit distance in real-world CBCD applications. One video could consist of one or several segments copied from different videos and, thus, copies may appear *locally*. For example, one common editing effect found on pirated web videos is to insert irrelevant frames at the beginning and the end of an original video. Another example is detecting commercials in a TV show. The query content, i.e. the commercials, might appear several times in the show. For either case, only a small portion of the query or the reference video is a copy. The second issue is the computational cost of computing the edit distance. Spurred on by the popularity of large video distribution web sites such as YouTube.com, a practical CBCD method must be highly efficient and scalable.

In this paper, we address both problems and propose an edit-distance-based approach that has the potential for large-scale CBCD applications. We first formulate a local alignment problem between two sequences and extend previous edit-distance-based approaches to compare video segments of *all* possible lengths. The main contribution of this work is the highly efficient matching process between a query video and a video database which is achieved by a fast local alignment method along with a dedicated index structure that provides detection acceleration at both the clip and frame levels. This method decomposes the design of the representation and matching, thus any frame-based representation could be easily incorporated into our framework. We demonstrate the effectiveness and efficiency of this method using the MUSCLE VCD benchmark [9]. We additionally suggest an application of the proposed method for fast shape retrieval.

In the remainder of the paper, we first provide background for the edit-distance-based approaches and present our implementation of various video signature and matching approaches. Section 3 describes our proposed method for efficient detection. Section 4 describes the experimental results for video copy detection. We also show results for fast shape retrieval using the same framework. Finally, we conclude the paper with a summary of our contributions and propose ideas for future research based on these concepts.

2. BACKGROUND

The process of pair-wise comparison of two videos is a fundamental task for determining whether one video contains sequences that are copies of sequences in the other. We start by describing the particular use of the edit distance for matching two videos. Approaches of this type differ from each other mostly in their sequence representation and their assignment of operation costs. In the following, we discuss the design options and our implementation.

2.1 Representation

2.1.1 Frame Sampling

For edit-distance-based approaches, the first step is to partition a video into a sequence of frames. To avoid unnecessary and duplicate comparisons for all frames, two sampling strategies are commonly used. First, a video is viewed as a list of shots represented by keyframes. This mapping of video to keyframes reduces the number of frames that needs to be analyzed by a factor of 100 to 5000, depending on the video content [20]. Although methods for detecting keyframes are, in general, quite robust for videos with the same format, different keyframe sequences might be generated when these techniques are applied to near-duplicate videos [20].

The second strategy is to sample frames at a fixed rate. This approach is simpler compared to automatic keyframe detection. In our experiments, we sampled one frame per second. Note that a sampled video sequence could still be long: a two-hour film, for example, has 7200 sampled frames.

2.1.2 Frame Description

In the second step, content in a frame is summarized by feature descriptors. Global statistics, such as color histograms, have been well developed and used successfully for content-based image retrieval. Bertini *et al.* proposed to use MPEG-7 descriptors—specifically the color layout, the edge histogram, and the scalable color descriptors—for effective video-clip matching [3]. Chum *et al.* showed that a color histogram, combined with a spatial pyramid placed over an image, could serve as a compact and discriminative descriptor for near identical image and shot detection [5].

Alternatively, local statistics, such as interest points with PCA-SIFT, have been applied in [5, 7, 17, 23]. This type of description is somewhat invariant, thus is highly robust, to image transformations such as occlusions and cropping. However, since one image can have hundreds to thousands of local features, matching between descriptors is computationally expensive. Although fast indexing structures (e.g. LSH) could help filter unnecessary comparisons among dissimilar features, matching based on local descriptors is still costly. Moreover, as indicated in [10], interest point detection alone is one of the computational bottlenecks in these methods. For videos, performing interest point detection on every extracted frame of a query video is simply infeasible due to the unacceptably high computational cost.

In general, global features are efficient to compute, compact in storage, but insufficiently accurate in terms of retrieval quality. Local features are more robust but computationally more complex and require more storage. A good tradeoff might be derived by the use of a semi-global descriptor. In this work, we extended the Markov stationary feature (MSF), which was proposed in [12] and has been shown to be effective for the task of TRECVID video concept detection. The MSF extends the histogram-based features by characterizing the *spatial co-occurrence* of histogram patterns using the Markov chain models. It therefore encodes spatial structure information both within and between histogram bins. We implemented the MSF-Color feature—the MSF extension of color histograms—in our experiments. To further enhance the feature by incorporating local information, we partition a frame into four regions, each of which is described using the MSF-Color feature. We quantized the HSV color space into 36 bins, resulting in a 288-dimensional (4 regions, 36 bins, 2 distributions) compact feature vector for each frame. Figure 1 summarizes the feature extraction process. Although the frame descriptor is based on appearance alone, the temporal aspects of a video are implicitly considered using the edit distance, which will be discussed in the next subsection.

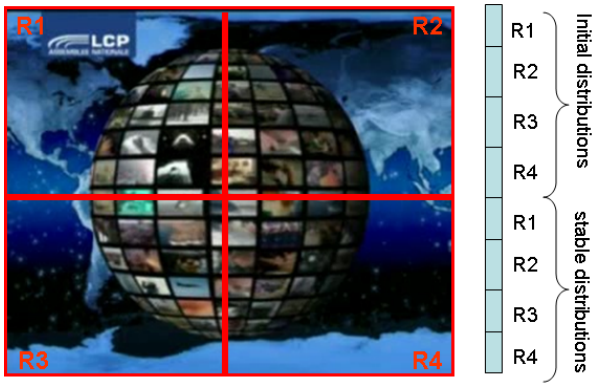

Figure 1. Extended MSF features. The left side shows the spatial division of a frame and the right side is the derived feature descriptor.

2.2 Matching Two Sequences

Motivated by the approximate string matching techniques, the use of edit distance in the context of video matching was first proposed in [1]. Given two strings X = [x_1, x_2, …, x_m] and Y = [y_1, y_2, …, y_n], over an alphabet $\sum$, and a set of operations (e.g. insertion, deletion, and substitution), the edit distance between X and Y is the minimal cost of applying a sequence of operations that transforms X into Y.

To use the edit distance for comparing two frame sequences, early studies suggested a quantization step that maps frames into symbols [1, 8]. For such cases, all operations have equal costs. The edit distance is then calculated as the number of operations needed to make two sequences of symbols equal. In [8], a heuristic method was proposed to determine the best step-size for quantization. However, the quantization process has potential drawbacks. It is not clear, for example, how one should choose the number of symbols for generic videos. Bertini *et al.* avoided this quantization step and proposed relying directly on the ground distance for comparing two frame descriptors to determine the operation costs [3].

The edit distance is flexible and can be easily adapted to applications by assigning different operation costs. For example, the longest common subsequence (LCS) matching technique used for measuring the similarity between video clips in [8] is a special case of using the edit distance for sequence matching. This technique allows insertion and deletion operations only, each of which incurs a unit cost. However, for some tasks such as identifying the appearance of a summarized video in a long sequence, deletions should be considered less expensive than insertions. The cost functions therefore need to be modified for such tasks.

Another view of the edit distance is that it measures how two sequences are *globally aligned*. By global we mean two sequences are aligned across their entire lengths. However, in many CBCD applications, the query sequence might not be a single video clip, but the concatenation of a collection of clips. Therefore, we often are more interested in finding the most *similar subsequences* within two sequences that are aligned pair-wise in the subsequence-level rather than finding the best alignment for the entire length of two sequences. Local alignment methods can return more than one match for subsequences among the two sequences under comparison because there may exist multiple-to-one, one-to-multiple, or multiple-to-multiple matches of subsequences. Therefore, for general CBCD applications, a metric based on local alignment is desirable.

3. FAST VIDEO COPY DETECTION

We extend the edit distance for the purpose of finding local alignments between two video sequences. We further propose a two-step method for accelerating the task.

3.1 Local Alignment

We extend the edit distance in two aspects to find the optimal local similarity. First, we derive a score $v(x_i, y_j)$ between two frame descriptors x_i and y_j based on their distance under the principle that $v(x_i, y_j)$ would be positive if x_i and y_j are similar, and negative otherwise. The value $v(x_i, y_j)$ can then be treated as the substitution "score." Moreover, we assign negative scores to insertions, denoted as $v(x_i, \varepsilon)$, and deletions denoted as $v(\varepsilon, y_i)$. Then, the optimal local alignment can be computed by dynamic programming that is very similar to the edit distance computation:

$$S(i,j) = \max\{0, \; S(i-1,j) + v(x_i,\varepsilon), \; S(i,j-1) + v(\varepsilon,y_j), \; S(i-1,j-1) + v(x_i,y_j)\}. \quad (1)$$

This is known as the Smith-Waterman algorithm [18]. The computational complexity is O(*mn*) and the storage is O(min(*m*, *n*)), where *m* and *n* are the lengths of the sequences under comparison. The local alignment is obtained by searching for the maximal score in the dynamic programming graph, and by tracing back the optimal path until a score of zero is retrieved. We use a simple linear model $v(x_i, y_j) = c - d(x_i, y_j)$ to derive the substitution score, where *c* is a constant and $d(x_i, y_j)$ is the χ^2 statistics between two normalized MSF-Color feature descriptors.

3.2 Acceleration

Filtration is a widely used technique for speeding up object detection or information retrieval tasks. For example, the popular face detection algorithm proposed by Viola and Jones [19] applies a cascade classifier structure that can quickly reject non-face regions so more computational resources can be reserved for examining more promising face-like regions. In information retrieval, the database is indexed so that when searching for similar instances for a query, only a small fraction of the dataset in the database needs to be examined. In this work, we apply filtration at two levels: 1) selecting candidate videos in the database, and 2) selecting good starting points for aligning two sequences.

3.2.1 Indexing Structure

We propose to use a vocabulary tree [13] for indexing all the frames extracted from the video database. The vocabulary tree was initially proposed for efficient image retrieval. In this work, each of the frame descriptors is hierarchically quantized in the vocabulary tree by hierarchical *k*-means clustering. Here, *k* defines the branch factor of the tree rather than the final number of clusters. The vocabulary tree allows a large and more

discriminative vocabulary to be used efficiently. It was shown experimentally in [13] that this indexing structure leads to a dramatic improvement in retrieval quality. Similar to the implementation in [13], we keep an inverted file associated with each leaf node—a representative frame (visual word)—in the vocabulary tree. However, we record not only the videos that contain frames of that word, but also those frame IDs. Intuitively, videos that have frames similar to the query should potentially be copies. Moreover, those similar frame pairs are candidate starting points for an alignment.

To improve the vocabulary tree for those tasks where videos can be added or removed from an active set over time, we implemented the adaptive vocabulary tree [22]. As its name implies, it adapts as instances are added to or removed from the database. One merit of the adaptive vocabulary tree is that we do not need to re-build the tree when the database slightly changes. In other words, an adaptive tree can be built incrementally. Moreover, the tree grows based on a measure that encourages splitting those nodes that become too ambiguous and pruning nodes that are not active for the current set of tasks. Thus, the distribution of descriptors decides the structure of the tree, which is also an important factor in determining the vocabulary's quality. Finally, the retrieval performance is less sensitive to the parameters—the number of branches and the capacity of a node—, as shown in [22]. As shown in Fig. 2, this indexing structure enables the retrieval time to grow sub-linearly with respect to the number of videos in the database.

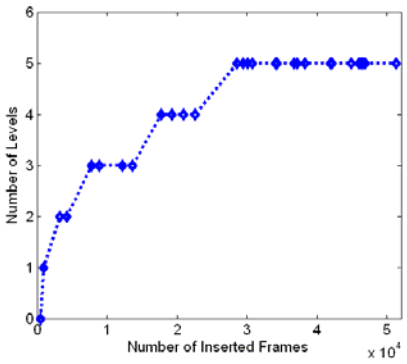

Figure 2. The levels of an adaptive vocabulary tree grow sub-linearly in terms of frame numbers in the database.

3.2.2 Fast Matching

The second opportunity for acceleration is to filter unnecessary alignments that would not possibly lead to successful matching. Suppose two sequences are unrelated; then, the best local alignment is no better than no alignment! Therefore, we can formulate an easier problem from the start: given two sequences, find those alignments that have a similarity exceeding a given threshold.

Inspired by FASTA [14], a fast algorithm used in bioinformatics for finding similar DNA and protein sequences, we first create a visual method called a *dot plot*. A dot plot puts a dot at (i, j) in an m by n matrix if the similarity between descriptor i and descriptor j exceeds a specific threshold. Figure 3 shows an example of the dot plot. This plot can be easily constructed by using those inverted files built in the previous step. Note that the dot plot is sparse if two videos under comparison are either completely or partially unrelated.

Figure 4 illustrates our search strategy, which consists of four steps. First, we identify all diagonals in the dot plot. A diagonal, shown as a diagonal line in Fig. 4 (a), represents consecutive matched frames of two video sequences. Next, those diagonals whose length is shorter than a pre-set threshold are discarded. That is, we filter out those single-frame matches and short aligned segments. This is illustrated in Fig. 4 (b). We then calculate a score, based on the *ground distance* between those frame descriptors that are aligned, for each remaining segment. Those segments with the top N highest scores are selected for further examination (Fig. 4 (c)). Among the selected segments, we try to join those that are close to each other in the dot plot with the goal of extending the overall length of the alignment. In this step, insertions and deletions are allowed, but there would be a penalty applied for such operations when connecting neighboring diagonals to form a longer segment. We obtain an approximate final score for each linked, longer segment by accumulating the scores of each of the connected diagonals minus those penalties caused by gaps between the diagonals. We return local alignments whose final scores exceed a threshold. The returned segments that consist of connected diagonals are illustrated in the gray areas in Fig. 4 (d). If the precise score for a linked segment is required, the Smith-Waterman algorithm can be applied. But due to its high computational complexity, it can only be applied to a much more restricted area.

The Smith-Waterman algorithm compares each frame of the query to every frame in the video database. Suppose the length of a query is m, and the size of the database (i.e. the number of frames) is N. The time complexity of the query would be $O(mN)$. In our fast method, we first construct dot plots by retrieving the corresponding visual word and its video and frame IDs for each query frame. This step takes $O(mL)$ using the vocabulary tree, where L is the depth of the tree. The complexity of deriving the local alignment from the dot plot depends on the number of dots in the plot. Suppose the size of the dot plot is m by n and it consists of r dots. Identifying diagonals requires a linear time $O(m+n)$, and the remaining processes for examining diagonals would take $O(r)$, overall. Since, in practice, dots are distributed sparsely and most dots can be eliminated in the initial filtration process, the overall runtime is generally linear, rather than quadratic, with respect to the sequence lengths.

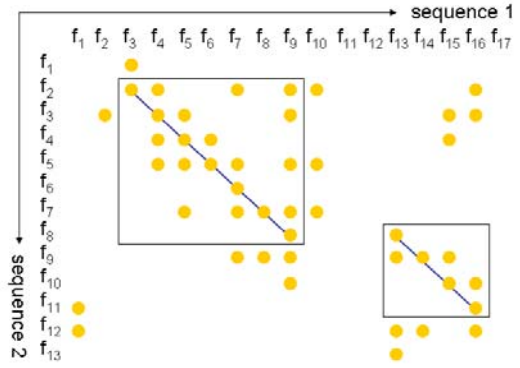

Figure 3. An example of the dot plot. Two sequences are locally aligned where the diagonals in the boxes indicate the region of alignment. See text for details.

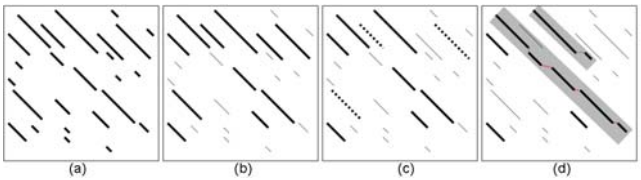

Figure 4. The search strategy: (a) determining diagonals, (b) eliminating short diagonals, (c) selecting the top *N* diagonals with the highest scores, and (d) linking diagonals and calculating the final score for each connected diagonal.

4. EXPERIMENTS

4.1 Dataset

We conducted experiments using the MUSCLE VCD benchmark [9]. This publicly available benchmark provides ground truth data for evaluating a system's detection accuracy based on two tasks: finding copies (ST1) and finding extracts (ST2). The first task evaluates a system's ability to find copies of whole videos in the database. This corresponds to the global alignment problem addressed in our framework. The second task is to detect regions of copies in the query, which is a local alignment problem. Both tasks are challenging because the transformations applied to this benchmark were very diverse.

This database consists of 101 videos with a total length of 80 hours. These videos come from different sources—web video clips, TV archives, movies—and cover various program types including documentaries, movies, sporting events, TV shows, and cartoons. Also, the videos in this dataset have different bit-rates, resolutions, and video formats.

4.2 Detection Results

We first show the performance of our recipe that uses the MSF-Color descriptor and the sequence matching method. Table 1 summarizes the results. Although the MSF-Color descriptor is a global feature [1], combining it with the sequence matching techniques surprisingly achieves good performance in comparison with other methods. Figure 5 shows two videos that failed in the ST1 task. It is apparent that the colors are changed substantially from the reference videos, and the magnitude of the change is greater than the range that the feature can cover. Poullot *et al.* has recently proposed a frame-level bag-of-features like description, and reported a score of 0.93 and 0.86 respectively for the same tasks [15]. It would be interesting to incorporate this feature into our framework in future experiments.

Table 1: Accuracy on the MUSCLE VCD benchmark

Method	ST1 score	ST2 segment score
CIVR07 Teams	0.46 ~ 0.86	0.17 ~ 0.86
Ours (MSF-Color+Edit)	0.86	0.76

[1] We have tried other global features, including MPEG-7 descriptors and the ordinary features. However, none of them can achieve accuracy higher than 62% for this dataset.

Figure 5. Two query examples for which our method failed. Left: queries, right: references.

4.3 Runtime Results

A sequence matching approach before applying any acceleration technique compares all frames in the query to all frames in the video database. For example, in ST1 there are 199,590 frames extracted from 101 reference videos and 11,232 frames from 15 queries. The total number of frame-pairs for comparison is approximately 2,246 million! In the first step of our proposed approach, we employ a vocabulary tree to filter out those videos which have just few frames similar to those in the query. Because each query in ST1 is a whole video copy and its reference video in the database was always among the ones with most matched frames, we selected the top 10 matched videos in this step for further examination. However, even though we filtered out 91 videos for each query, there are still more than 341 million frame-pair computations remained.

The second step, as discussed in Section 3, uses the inverted files associated with tree nodes to derive a dot plot between two sequences. The dot plot is, in general, sparse. The density, defined as the number of dots divided by the total plot area, of those dot plots between each query in ST1 and the top 10 selected videos is, on average, 32.93%. The density ranges from 1.33% to 89.12%. Recall that fast matching is performed by first identifying diagonals on the dot plot, removing short diagonals, computing a score for each of the remaining diagonals, selecting those with top 10 highest scores, and finally linking them into longer diagonals. We conduct frame comparisons only if a pair is located on a sufficiently long diagonal. In this step, the threshold of the diagonal length for filtration is an important parameter for adjusting the tradeoff between accuracy and speed. A larger threshold reduces the number frame comparisons but increases the risk of missing segments that are true copies. Figure 6 shows the number of frame pairs that need to be compared versus the threshold of the diagonal length for ST1. Based on these results, we selected 1/16 of the query length for our experiment—it did not cause any detection rate drop and the comparison count is reduced by approximately 85%.

For ST2, since each copy only represents a small portion (3.5%-12.6%) of the query, the pyramid matching score would not be a good metric for measuring of the relevance between the query and the reference videos. Therefore, for this task we do not filter any

video and Step 1 is solely used for creating dot plots. Furthermore, we set an absolute threshold (15) of the diagonal length in our experiments. Figure 7 shows an example of the alignment, where white pixels denote original dots, red pixels denote diagonals with a length longer than the threshold, and green pixels denote the alignment found.

Table 2 summarizes the total runtime in seconds, which is based on non-optimized MATLAB codes running on a machine with a 2.4 GHz P4 CPU and 768 MB of RAM. Our method, with a runtime of about 1/4-1/8 of the query length, is a viable solution to applications that require real-time processing.

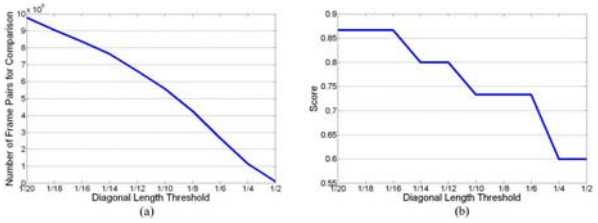

Figure 6. The diagonal length threshold vs. the performance: (a) number of frame pairs needed for comparison, which will be proportional to the speed, (b) ST1 score.

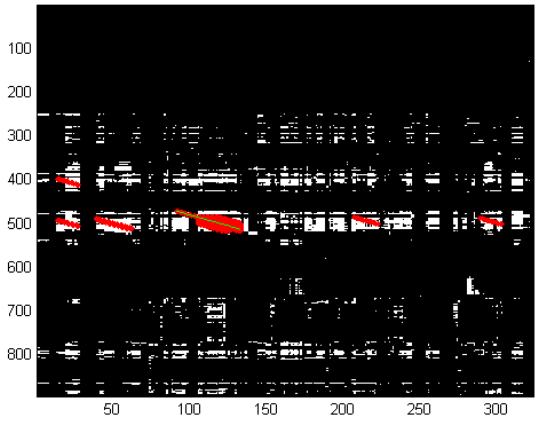

Figure 7. An example of local alignment in ST2.

Table 2. Total runtime (in seconds).

Task	Query length	Sequence matching	Ours	Speedup
ST1	11,232	26,307.63	1,394.61	18.86x
ST2	2,690	10,492.33	570.09	18.40x

4.4 Fast Shape Retrieval

The proposed fast matching method for efficient video copy detection can actually be applied to several other visual retrieval tasks. In this subsection, we demonstrate experimental results for its application to fast shape retrieval.

Shape matching techniques can be used as a basis for object category recognition or for hand-written text recognition [2]. In many state-of-the-art shape matching approaches, a contour image is described as a sequence of local descriptors gathered from the contour and the similarity between two sequences is measured by edit-distance-based metrics. These approaches are effective, but are difficult to apply for large-scale applications since they are computationally expensive.

Those approaches can be significantly accelerated using the proposed framework. We conducted experiments on the MPEG-7 shape database, namely the Core Experiment CE-Shape-1 part B, which measures the performance of similarity-based shape retrieval. The database consists of 1,400 shapes in 70 categories, each of which consists of 20 shapes. We followed the same setup used in [21] for our experiments. For each shape image, we uniformly sampled 100 points from the contour of each shape. Each point is described by the shape context descriptor [2]—a 60-dimensional feature vector—, which encodes the relative coordinates of remaining points using the log-polar space. To compare two shape images, a variant of the edit distance was used to search for optimal alignment and to derive a similarity score.

Before applying the acceleration technique, this process took 95.95 hours to complete. The search requires 3.92×10^{10} descriptor-pair comparisons. Our method first indexed all the shape context descriptors, resulting in a tree of 4,097 nodes and 5 levels. Since one shape is a global copy of others belonging to the same class, their similarities—which are based on the pyramid matching kernel—are very accurate. We selected the top 140 (1/10) candidate shapes in the first step and further examined them in the second step. The total processing time after the application of our acceleration technique is 6.23 hours—a 15.4X speed-up in comparison to the original sequence matching techniques and the same level of retrieval accuracy is achieved.

5. CONCLUSIONS

The edit distance is a powerful metric for measuring the dissimilarity between two video sequences and its variants can be used to effectively and efficiently identify video segments that are locally aligned. In this paper we formulate the video copy detection problem as a local alignment problem between video sequences. We propose a two-step method to speed up the edit-distance-based approaches which address the formulated problem. Results on the MUSCLE VCD benchmark and the MPEG-7 shape dataset demonstrate significant computational improvement without sacrificing accuracy.

One direction of our future research is to design a more effective feature descriptor. Frame representation is very crucial to the detection performance. Although we showed that a semi-global descriptor provides promising discriminative power, there is still room for improvement in comparison with those representations based on local features [15]. Since our method decomposes the representation and the indexing/matching process, any frame-based representation could be easily incorporated into our framework. Another direction is to explore multiple sequence alignment techniques to find essential content within multiple relevant video streams. This could be a useful tool for creating a summary from huge volumes of near-duplicate videos on video sharing websites.

6. REFERENCES

[1] D. A. Adjeroh, M. -C. Lee, and I. King. A distance measure for video sequence similarity matching. In *Proceedings of the International Workshop on Multi-Media Database Management Systems*, pages 72-79, 1998.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI'98)*, 24(4): 509-522, 2002.

[3] M. Bertini, A. D. Bimbo, and W. Nunziati. Video clip matching using MPEG-7 descriptors and edit distance. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07)*, pages 133-142, 2006.

[4] S. -C Cheung, and A. Zakhor. Fast similarity search and clustering of video sequences on the world-wide-web. *IEEE Transactions on Multimedia*, 7(3): 524-537, 2004.

[5] O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07)*, pages 549-556, 2007.

[6] A. Joly, O. Buisson, and C. Frelicot. Content-based copy retrieval using distortion-based probabilistic similarity search. *IEEE Transactions on Multimedia*, 9(2): 293-306, 2007.

[7] Y. Ke, R. Sukthankar, and L. Houston. Efficient near-duplicate detection and sub-image retrieval. In *Proceedings of the ACM International Conference on Multimedia (MM'04)*, pages 1150-1157, 2004.

[8] Y. Kim, and T. -S. Chua. Retrieval of news video using video sequence matching. In *Proceedings of the International Multimedia Modelling Conference (MMM'05)*, pages 68-75, 2005.

[9] J. Law-To, A. Joly, and N. Boujemaa. Muscle-VCD-2007: a live benchmark for video copy detection, 2007. http://www-rocq.inria.fr/imedia/civr-bench/.

[10] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust voting algorithm based on labels of behavior for video copy detection. In *Proceedings of the ACM International Conference on Multimedia (MM'06)*, pages 835-844, 2006.

[11] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07)*, pages 371-378, 2007.

[12] J. Li, W. Wu, T. Wang, and Y. Zhang. One step beyond histograms: Image representation using Markov stationary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1-8, 2008.

[13] D. Nister, and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 2161-2168, 2006.

[14] W. R. Pearson, and D. J. Lipman. Improved tools for biological sequence comparison. In *Proceedings of the National Academy of Sciences of the United States of America*, 85(8):2444-2448, 1988.

[15] S. Poullot, M. Crucianu, and O. Buisson. Scalable mining of large video databases using copy detection. In *Proceedings of the ACM International Conference on Multimedia (MM'08)*, pages 61-70, 2008.

[16] S. Poullot, O. Buisson, and M. Crucianu. Z-grid-based probabilistic retrieval for scaling up content-based copy detection. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07)*, pages 348-355, 2007.

[17] J. Sivic, and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'03)*, pages 1470-1477, 2003.

[18] T. F. Smith, and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1): 195-197, 1981.

[19] P. Viola, and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2), pages 137-154, 2004.

[20] X. Wu, A. G. Hauptmann, and C. -W. Ngo. Practical elimination of near-duplicates from web video search. In *Proceedings of the ACM International Conference on Multimedia (MM'07)*, pages 218-227, 2007.

[21] M. Yeh, and K. -T. Cheng. A string matching for visual retrieval and classification. In *Proceedings of the ACM International Conference on Multimedia Information Retrieval (MIR'08)*, pages 52-58, 2008.

[22] T. Yeh, J. Lee, and T. Darrell. Adaptive vocabulary forests for dynamic indexing and category learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'07)*, pages 1-8, 2007.

[23] D. -Q. Zhang, and S. -F. Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *Proceedings of the ACM International Conference on Multimedia (MM'04)*, pages 877-884, 2004.

[24] J. Zhou, X. -P. Zhang. Automatic identification of digital video based on shot-level sequence matching. In *Proceedings of the ACM International Conference on Multimedia (MM'05)*, pages 515-518, 2005.