Fast and Robust Search Method for Short Video Clips from Large Video Collection

Junsong Yuan [1,2], Qi Tian [1], Surendra Ranganath [2]
[1] Inst. for Infocomm Research, Singapore [2] Dept. of ECE, National Univ. of Singapore
Email: {jyuan, tian}@i2r.a-star.edu.sg, elesr@nus.edu.sg

Abstract

In this paper a fast and robust method is proposed to search a large video collection for given short clips. Compared with existing video searching methods which use visual features only, our scheme performs a two-phase hierarchical matching technique using visual and audio features successively. Considering that video sampling rate (25 or 30 fps) is much lower than that of audio (8 to 48 kHz), a coarse search is implemented with sub-sampled video frames first, and then potential matches will be verified and accurately located using fine audio features. Both features are extracted directly from MPEG compressed video for computational efficiency. Experiments have been conducted on over 10.5 hours of video to search for re-occurrences of 83 TV commercials and one news lead-out clip. All the 220 instances are correctly detected with no false alarm. Our experiments also show that the proposed method is robust to variations of video bit rate, frame rate, frame size and color shifting.

1. Introduction and related work

In this paper, a two-phase video searching method using visual and audio features successively is proposed, with the aim of achieving high search accuracy and low computation. Compared with other methods considering coarse visual features only, we incorporate fine audio features for verification and accurate localization.

To search given short clips from long video streams, the common method is to effectively represent the given clips first and then sequentially search for them in the target streams. Current video searching methods based on representative image matching can be summarized into three main categories: frame sequence matching [2]-[5], key-frame based shot matching [6]-[8] and sub-sampled frame matching [8]-[10].

Although frame sequence matching attained certain level of success in [2]-[4], the common drawback of these techniques is the heavy computational cost of the exhaustive search. [5] improved on this by skipping unnecessary steps during the search, while guaranteeing exactly the same search result as exhaustive search. However, we believe that the search speed can also be improved by using coarser visual features. Besides, methods in [5] need to be tested with different video variations.

Key-frame based shot matching is another popular method for video identification and retrieval [6] [7]. When applied to short clip searching, this method, however, has some drawbacks. First, the performance of shot representation strongly depends on the accuracy of shot segmentation and characteristics of the video content. For example, if the given clip has blurry shot boundaries or very limited number of shots, shot-based searching will not produce good results. Second, shot resolution, which could be a few seconds in duration, is usually too coarse to accurately locate the instances in the video stream.

Some other methods [8]-[10] consider sub-sampled frame matching for video stream searching. Although search speed can be accelerated by using coarser temporal resolution, these methods may suffer from inaccurate localization. And when the sub-sampled frames of the given clip and that of the matching window are not well aligned in temporal axis, it will affect the matching result. [10] partially overcomes this sub-sampled frame shifting problem and is robust to video frame rate change. However, feature extraction in [10] is time consuming, therefore not suitable for on-line processing and large database searching.

By being able to recognize and locate multiple occurrences of given video clips, such as commercials and news program lead-in & lead-out theme-music, in unknown video streams, one can verify advertisement air time and reveal the broadcast structure. Different from video content detection, such as commercial break detection [1] whose purpose is for commercial filtering, video stream searching seeks to identify re-occurrences of given individual clips and needs to accurately locate its position if an instance is found inside the long streams.

2. Feature extraction

Our feature extraction considers four principal challenges for video search: The feature should be compact while being informative enough for the identification task; the selected visual features must be robust to likely distor-

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

tions and alteration of the video content; the feature resolution must be efficient for fast search, but also fine enough to accurately locate the instances; and the feature extraction must be computationally efficient. In order to reduce computational complexity, we extract both audio and visual features directly from the compressed domain.

2.1. Visual feature extraction and representation

In our approach, we simply use I frames of the MPEG videos to characterize video content. Two main advantages are obtained by using I frames as representative images. Firstly, the uniformly sub-sampled I frames of MPEG video have coarse temporal granularity, which is typically hundreds of milliseconds depending on the MPEG GOP parameter, therefore I frames can compactly represent the video while still being informative enough for the recognition task. Secondly, the color features of I frame can be extracted directly from compressed video stream [4] to avoid decoding cost.

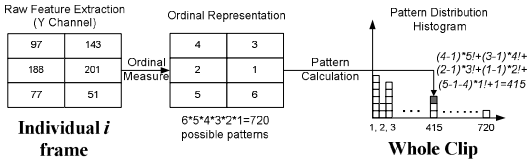

Figure 1. Compact visual features to represent the whole clip (take Y channel for instance).

As described in Figure 1, each I frame is represented by a reduced image, of size 2×3. For each Y, Cb or Cr channel, we calculate the average value of each of the 6 sub-images, using dc coefficients directly. After raw feature extraction, in total 18 (#Y/#Cb/#Cr=6/6/6) coefficients will represent an individual I frame.

Raw feature extraction is then followed by ordinal measure representation as in [3]. Since ordinal measure is non-sensitive to uniform color shifting, it can represent the video content robustly. Considering that the total combination of the ordinal measures is limited (6! =720 possible patterns for the given 6 sub-images), each possible ordinal measure combination can be treated as an individual pattern. Therefore for a set of key frames to represent a given clip or corresponding matching window in the video stream, we can form the normalized pattern distribution histogram over the whole video clip to represent it globally and compactly.

After the above operations, the clip can be represented by 3 normalized 720-dimensional histograms, corresponding to Y, Cb, and Cr channels respectively. For each channel, the clip is represented as:

$$H=(h_1,h_2,\cdots,h_i,\cdots,h_N) \quad 0\le h_i \le 1 \quad and \quad \sum_i h_i = 1$$

Here N=720 is the number of histogram bins, namely the number of possible patterns mentioned above.

The advantages of using ordinal measure based histograms as visual features are two fold. First, they are robust to frame size change and color shifting problem as mentioned above. Secondly, the shape of the pattern distribution histogram can describe the whole clip globally; therefore it is insensitive to video frame rate change and other local frame changes compared with frame sequence matching in [2]-[5].

2.2. Audio feature extraction

Since coarse visual feature may cause potential false matching and can't accurately locate the instance, fine audio features are introduced in this section in order to locate the instance more accurately and also for verification.

From the available audio features, we choose audio loudness and SFM (spectrum flatness measure) features, which have demonstrated good performance in [12]. In our method, both features are estimated directly from 32 sub-band coefficients of the compressed audio granules without decompression. In MPEG 1 layer 1/2 audio compression, each non-overlapped compressed audio granule represents a temporal window that contains 384 audio samples. Both Loudness and SFM features were calculated based on 4 frequency bands with equal length. Therefore in total 8 coefficients (#Loud/#SFM=4/4) will represent an individual compressed audio granule, instead of its 32 sub-bands coefficients. Finally an $n \times 8$ matrix will be generated to represent the whole clip, where n is the clip length, namely the number of granules.

3. Hierarchical video search algorithm

3.1. Two-phase sequential search

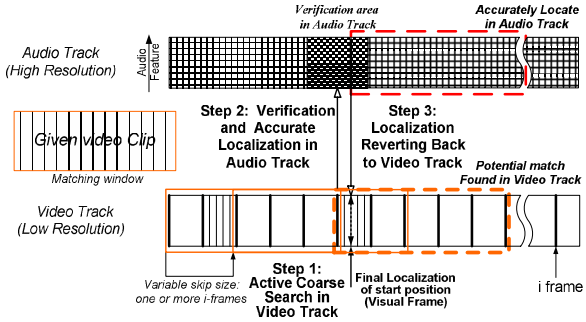

Figure 2. Two-phase sequential video search.

The two-phase sequential localization process is described in Figure 2. In the first phase, sequential search will go through the target stream based only on visual features described in Section 2.1. Search granularity in this phase is the interval between two I frames. If a potential match is detected, then audio features will be extracted from the audio track, as described in Section 2.2, to search around the potential position corresponding to the result from video track, for verification and accurate localization. For robustness, the verification area of the audio track is extended to one more neighboring I frame on each side of the given position. In the last step, the start position of the found instance will be converted from audio track back to video track, finally locating the start position to a video frame.

3.2. Coarse visual match

For visual feature histogram matching, we use the reciprocal of Euclidean distance as the similarity measure between the given clip H_v and the sliding matching window H_{sw}. For each color channel, similarity is defined as:

$$S_c(H_v, H_{sw}) = \frac{1}{\sqrt{\sum_{i=1}^{N}(H_v(i) - H_{sw}(i))^2}} \qquad c = Y, Cb, Cr$$

The similarity over the whole clip is defined as the maximum similarity value of three Y, Cb, and Cr channels:

$$S(H_v, H_{sw}) = \max\{S_c(H_v, H_{sw})\} \qquad c = Y, Cb, Cr$$

Let the similarity measure array be $\{S_i; 1 \le i \le m+n-1\}$ corresponding to *m+n-1* sliding windows, where *n* and *m* are the key frame number of given clip and target stream respectively. Based on [5] and [11], the search process can be accelerated by skipping unnecessary w_i steps.

$$w_i = \begin{cases} floor\,(\sqrt{2}D(\frac{1}{S_i} - \theta)) + 1 & if\ S_i < \frac{1}{\theta} \\ 1 & otherwise \end{cases}$$

where D is the number of key frames of the corresponding matching window. θ is the predefined skip threshold.

Similar to [5], potential start position of the match will be determined by local maximum above the threshold, which fulfills the following conditions:

$$S_{k-1} \le S_k \ge S_{k+1} \quad and \quad S_k > \max\{T, m + k\sigma\}$$

where T is the pre-defined preliminary threshold, m is the mean and σ is the deviation of the similarity curve; k is an empirically determined constant. Only when similarity value exceeds the maximum value of T and $m + k\sigma$, can it be treated as a potential match position.

3.3. Fine audio match

Fine audio matching has two fold meaning here. First, compared with compact visual features, we use an $n \times 8$ matrix as audio features, which could represent the whole clip more effectively. Secondly, the finer temporal resolution of audio features makes possible finer location granularity in audio track. Compared with visual search steps which can be hundreds even thousands of milliseconds using skip strategy, search step in the audio track is only tens of milliseconds, and can thus guarantee accurate localization.

The distance that measures the dissimilarity of the search clip A and the corresponding matching window SW in the target audio track is defined as below, both A and SW are represented by $n \times 8$ feature matrixes; n is the number of compressed audio granules in the matching window; the search granularity is set to the audio feature resolution:

$$D(SW, A) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{8}\left|SW_{ij} - A_{ij}\right|$$

The final location D_k of the instance in audio track is decided by the following conditions:

$$D_{k-1} \ge D_k \le D_{k+1} \quad and \quad D_k < \theta_{\text{audio}}$$

where θ_{audio} is the predefined fixed detection threshold.

4. Experiment

All the simulations were on a standard P4 @ 2.53G Hz PC (512 M memory). The algorithm was implemented in C++. The clip set consists of 83 individual commercials which varied in length from 5 to 60 seconds (Figure 3) and 1 10-second long news program lead-out clip (Figure 4). All the 84 given clips were taken from ABC TV news programs. The experiment seeks to identify and locate the instances of these clips inside the target video collection, which contains 22 half-hour long ABC TV news broadcasts. The 83 commercials appear in 209 instances in these half-hour news programs; and the lead-out clip appears in 11 instances. All the video data were encoded in MPEG1 at 1.5 Mb/sec with image size of 352×240 or 352×264 and frame rate of 29.97 fps. It is compressed with the frame pattern IBBPBBPBBPBB, with I frame resolution around *400ms*. The associated audio track is encoded into MPEG1 layer 2 at 192kb/sec with sampling rate at 32 kHz, and compressed audio granule resolution *12ms* (384 audio samples).

Figure 3. Sample of ABC commercial clip.

Figure 4. ABC news program lead-out clip.

The performance of proposed algorithm is evaluated as below: for each of the 84 given clips, we search through all the 22 target streams. Therefore $84 \times 22 = 1848$ similarity curves were generated after search. Based on the detection criteria in Section 3.2 and 3.3, automatic detecting will go through these curves. After that, detection results will compare with the ground truth, which is manually obtained. The performance is finally measured by precision and recall defined below.

(*Precision = detects / (detects + false alarms)*)
(*Recall = detects / (detects + miss detects)*)

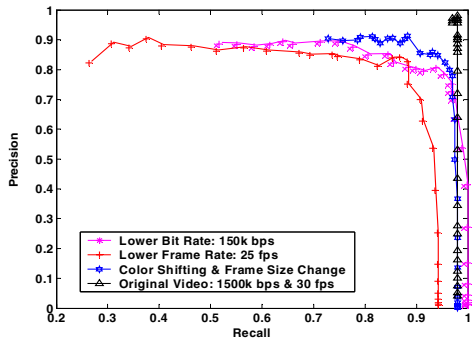

Figure 5. Robust test result using visual feature only.

In order to investigate the robustness of our method, we also tested the same 10.5 hour video collection set with different video variations and search for the original given clips back to these streams. The result is shown in Figure 5; note **only** visual feature is used. The detection curves are based on varying visual detection threshold, namely varying parameter k in Section 3.2. From the result we can see that the presented algorithm is robust to different video changes, such as lower frame rate and bit rate, frame size change and color shifting. If using visual-audio features together, detection result in Figure 5 can be improved to obtain 100% precision by audio verification, and leave the recall rate the same.

Table 1. Result of instance locating accuracy

Average localization error	
Coarse Visual Features Only (sub-sampled I frames)	Audio – Visual Features
550 ms	**60 ms**

Table 1 gives the location accuracy of the proposed method. We found that when using visual-audio features together, all of the given clips can be accurately located to the correct position in audio track, with localization granularity of 12 ms. The localization error, which can be up to 5 video frames for some typical commercials, mainly comes from the audio-video non-synchronization problem, namely the third step in Figure 2. Therefore although instances can be accurately located in audio track (with 12 ms granularity), because of the synchronization delay, localization shift will occur when reverting the position back to video track. On the average the localization error is around ***2*** video frames.

Table 2. Computational cost comparison

Task [Video Collection: 10.5h MPEG1 video / 15 sec given clip]	**CPU Time Cost (sec)**	
	NTT	Proposed
Visual Feature Extraction	video decoding cost + 1577.1	1106.5
Local Audio Feature Extraction		57.6
Visual Active Search	0.37	0.15
Audio Fine Localization		0.37
Total Search Cost	0.37	0.52

Table 2 presents a preliminary computational cost comparison of NTT's method [5] and our method. The search cost of NTT's active search is estimated under the assumption that average skip step of active search is 20 video frames. It can be seen that feature extraction cost has been largely reduced by extracting features directly from compressed domain. The two-phase search is also fast as NTT's active search, and could accurately locate the instances. This is because large computation has been saved by introducing coarse visual search in the first phase. And only when potential match is found in the first phase, is further fine audio matching needed.

5. Conclusion and Future Work

We have presented an algorithm for detecting reoccurring instances of given short clips in large video collection. Compared with existing methods, this algorithm has solved the video clip detection problem more efficiently by combining visual and audio features. Different from NTT's active search method, which mainly depends on its heuristic skip strategy to speed up search process, our scheme properly makes use of the different characteristics of audio and video sampling rate to accelerate search speed, which is also very fast and efficient. The experiments show the proposed search method can achieve 100% accuracy in detecting given short video clips from the video collection encoded with the same coding parameters as used for the given short video clips, otherwise about 90% detection performance can be achieved. Due to the fine sampling rate of audio, the proposed method can locate multiple copies of given short video clips from large video collection with very high temporal accuracies: accurate to one or a few video frames.

Besides commercials, the proposed scheme can also be extended to search for arbitrary short video clips. The applications include Google-like video search, TV program structure analysis where occurrences of the lead-in & lead-out video clips can be accurately located, including sports, station logo, weather report, financial news, etc., and content copy management. Our future works include to improve the robustness of the detection to video and audio encoding variations and to further reduce the computational cost.

5. Reference

1. L. Agnihotri et al., "Evolvable Visual Commercial Detector," In *Proc. of CVPR'03*, Vol. 2, pp. 79-84, 2003
2. R. Lienhart, et al., "On the Detection and Recognition of Television Commercials," In *Proc. IEEE Conf. on Multimedia Computing and Systems*, pp. 509-516, 1997
3. Rakesh Mohan, "Video Sequence Matching," In *Proc. of ICASSP'98*, Vol. 6, pp. 3679-3700, 1998
4. M.R. Naphade, et al., "A Novel Scheme for Fast and Efficient Video Sequence Matching Using Compact Signatures," In *Proc. SPIE, Storage and Retrieval for Media Databases 2000*, Vol. 3972, pp. 564-572, 2000
5. K. Kashino et al., "A Quick Search Method for Audio and Video Signals Based on Histogram Pruning," In *IEEE Trans. on Multimedia,* Vol. 5, No. 3, pp. 348-357, 2003
6. Juan M. S. et al., "Shot Partitioning Based Recognition of TV Commercials," In *IEEE Trans. on Multimedia Tools and Applications*, vol. 18, pp. 233-247, 2002
7. T. C. Hoad et al., "Video Similarity Detection for Digital Rights Management," In *Twenty-Sixth Australasian Computer Science Conference*, Vol. 16, 2003
8. A. K. Jain et al., " Query by video clip," In *Multimedia System*, Vol. 7, pp. 369-384, 1999
9. S. Kim et al., "An efficient algorithm for video sequence matching using the modified Hausdorff distance and the directed divergence," In *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, pp. 592-596, 2002
10. Nicholas Diakopoulos et al., "Temporally Tolerant Video Matching", In *SIGIR Multimedia Information Retrieval Workshop 2003*, Toronto, Canada, Aug. 2003
11. A. Kimura, et al., "A Quick Search Method for Multimedia Signals Using Feature Compression Based on Piecewise Linear Maps," In *Proc. of ICASSP'02*, Vol. 4 , pp. 3656 -3659, May 2002
12. E. Allamanche, et al., "Content-Based Identification of Audio Material Using MPEG-7 Low Level Description," In *Proc. of the International Symposium of Music Information Retrieval*, Bloommington, Oct. 2001