Fast coarse-to-fine video retrieval via shot-level statistics

Yu-Hsuan Ho[a], Chia-Wen Lin[1a], Jing-Fung Chen[b], and Hong-Yuan Mark Liao[b]

[a]Dept. Computer Science & Information Eng., National Chung Cheng Univ., Chiayi 621, Taiwan
[b]Institute of Information Science, Academia Sinica, Taipei 115, Taiwan

ABSTRACT

We propose a fast coarse-to-fine video retrieval scheme using shot-level spatio-temporal statistics. The proposed scheme consists of a two-step coarse search and a fine search. At the coarse-search stage, the shot-level motion and color distributions are computed as the spatio-temporal features for shot matching. The first-pass coarse search uses the shot-level global statistics to cut down the size of the search space drastically. By adding an adjacent shot of the first query shot, the second-pass coarse-search introduces the “causality” relation between two consecutive shots to improve the search accuracy. As a result, the final fine-search step based on local color features of key-frames of the query shot is performed to further refine the search result. Experimental results show that the proposed methods can achieve good retrieval performance with a much reduced complexity compared to single-pass methods.

Keywords: Video retrieval, query by clip, video matching, coarse-to-fine search, video database

1. INTRODUCTION

Due to the popularity of the Internet and the powerful computing capability of computers, efficient processing/retrieval of multimedia data has become an important issue. A great number of researchers, no matter from academia or industry, have contributed their effort on developing core technologies for multimedia processing, transmission, storage, and retrieval[1-9]. Multimedia is a general term for different media. It can be video, audio, graphics, text, image, or the combination of any two or more of the above mentioned media. Among different types of media, video contains the most amount of data and is relatively hard to be dealt with due to its complexity. In order to efficiently manage video data, including how to perform appropriate indexing[1-12], efficient storage and transmission[13-15,17,18] and fast retrieval[6,11,12,25], one has to put his/her effort on developing better video compression, indexing, and fast search algorithms. For the purpose of fast transmission, the MPEG video compression techniques have been well developed. However, the compressed video files still have a huge amount of data and are still far away from the goal of efficient retrieval. In order for executing efficient video indexing/retrieval, many crucial technologies such as shot change detection [21,22], shot representation[4,5], key video frame/clip extraction[7-10,13], video matching metrics[11,12], and video summarization techniques[7,13-15], have been developed in the past decade. The objective of the above mentioned technologies is to effectively reduce video data amount to the smallest extent and at the same time maintain the original meaning of a video. The most basic unit of a video that holds semantic meaning is the so-called shot. A shot is the collection of successive frames in a video, and is purely described by a continuous action in time or space[21,22]. After all shot boundaries are detected, for the purpose of fast retrieval, several crucial issues have to be tackled. These issues include: (1) how to group the shots that are spatially close to each other and similar in contents; (2) how to annotate a shot so that the subsequent retrieval task can be facilitated; and (3) how to design an algorithm to perform efficient video retrieval. In this paper, we intend to propose a powerful video retrieval scheme to tackle the above mentioned issues.

In the multimedia era, using an unknown video clip to retrieve a complete counterpart video in a video database will definitely be the future trend of our daily life. The video retrieval scheme proposed in this paper is a coarse-to-fine shot-based approach. In general, a video clip may contain several consecutive shots and its first and last shots are most of the time incomplete. For correctly detecting-shot boundaries, we use two gradual shot change detection algorithms[21,22] developed by ourselves to achieve the goal. Fig. 1 depicts the block diagram of the proposed coarse-to-fine video retrieval scheme, which consists of a two-step coarse search and a fine search. The objective of the coarse-search is to select a reasonable number of candidate video clips from a video database while avoiding the miss detection of correct

[1] cwlin@cs.ccu.edu.tw; phone +886 5 272 0411 ext. 33120; fax +886 5 272 0859; http://www.cs.ccu.edu.tw/~cwlin

clips. At the coarse-search stage, we compute the entropy of the motion vectors from every constituent shot of a query video clip and then pick the shot that has the maximum entropy as the query shot. The reason of choosing the shot that has the maximum entropy is due to its most discriminating power that can be applied to conduct an efficient search. The first-step coarse-search identifies some similar video clips by using some shot-level spatio-temporal statistics (i.e., the motion and color histograms). The above process can significantly cut down the size of the search space. Subsequently, in the second-step coarse-search, a shot that is adjacent to the first query shot (either preceding or subsequent) is chosen, and the two consecutive shots are concatenated to form a two-shot query. The motion and color features used are the same as those used in the first step. However, the "causality" relation that confines the order of two consecutive shots is introduced to strengthen the discriminating capability. In the coarse-search process, we extract the object motions of a query shot and then quantize them into the form of 2-D probability distributions. The feature of this form is the temporal feature that will be used in our scheme. The color histogram of the same shot, on the other hand, will be used as the spatial feature. For matching two shots with different lengths, their corresponding motion and color probability distributions are compared by the discrete Bhattacharyya distance[16] which is designed specifically for comparing two arbitrary distributions. As a result, the joint distance which sums up the distance of the motion statistics and that of the color histograms is used to measure the similarity between the two shots.

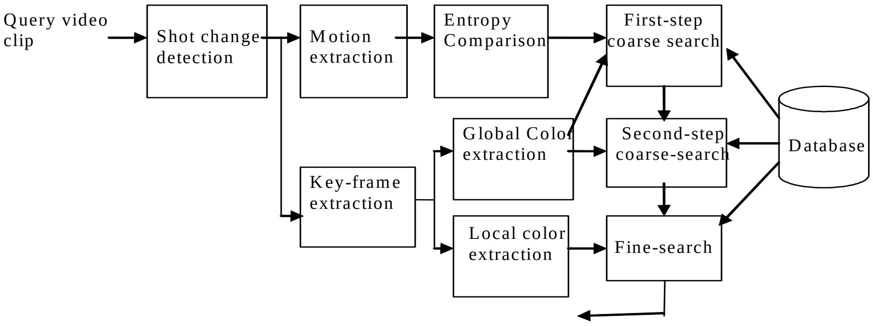


Fig. 1. Block diagram of our proposed multi-pass coarse-to-fine video retrieval scheme.

To further reduce the search space we propose to perform a fine-search after the two-step coarse search. In the fine search process, we extract color features from a set of selected key-frames and then use them to further reduce the search space. We divide each selected key-frame into 2×2 subimages and the local color histogram of each subimage is calculated individually. Then, we calculate the Bhattacharyya distance and use it to choose the closest shots from the database. In order to save the computation time in the fine search process, we only compute the color histogram of the key-frames rather than computing the color histograms of all frames in a shot. Experimental results have proven that our proposed video retrieval scheme is indeed powerful in terms of efficiency as well as accuracy.

The rest of this paper is organized as follows. Section 2 describes the proposed two-step coarse-search scheme using the shot-level statistics and the causality factor. Section 3 describes the fine-search scheme which utilizes the color distributions of key-frames. Section 4 shows the experimental results. Concluding remarks will be drawn in Section 5.

2. COARSE SEARCH USING SHOT-LEVEL STATISTICS AND CAUSALITY

2.1 Pre-processing

Since a shot is the most primitive unit with semantic meaning that can be used for video retrieval, in our method, each video clip is segmented into its constituent video shots by using our previous methods[21,22]. A complete shot may consist of several GOPs (group of pictures). Under the circumstances, the original design of an MPEG bitstream cannot be used directly for computing the local motions (object motions) between every consecutive anchor frame pair (anchor frames mean I- and P-frames). For a P-frame in an MPEG bitstream, the forward motion vectors between itself and its reference frame can be directly extracted and used as a valid motion field. However, it is not the case for an I-frame because an I-frame is actually intra-coded and contains no motion information. For the above mentioned discontinuity problem that does exist between two consecutive GOPs, the motion vectors of the B-frame which is located right after the last P-frame in each GOP are used to solve the problem. Suppose the forward motion vector and backward motion vector of a macroblock in the above mentioned B-frame are *F* and *B*, respectively. Then *F* - *B* can be regarded as a pseudo motion vector between the last P-frame in the current GOP and the I-frame in the next GOP.

In order to extract local object motions as a feature for video retrieval, the global motion caused by camera operations need to be excluded. We use the following four-parameter affine motion model[24] to characterize the camera motion.

$$\mathbf{mv}_{\text{cam}} = \begin{bmatrix} zoom & rotate \\ -rotate & zoom \end{bmatrix} \cdot \begin{bmatrix} mx_x \\ mx_y \end{bmatrix} + \begin{bmatrix} pan \\ tilt \end{bmatrix} \tag{1}$$

Using the camera motions derived by the above model, the local motion corresponding to each macroblock sequence are estimated correctly by subtracting the original motion vectors by the estimated camera motions.

Since a local motion vector derived from two consecutive macroblocks in a shot may be large in magnitude, we have to transform it into a smaller domain. To reduce the dynamic range, a motion vector (mv_x,mv_y) is transformed into the *UV* plane by the following quantization operation.

$$\mathbf{qmv} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \lfloor mv_x / I + 0.5 \rfloor \\ \lfloor mv_y / I + 0.5 \rfloor \end{bmatrix} \tag{2}$$

where *I* is an integer that can be used to control the degree of quantization.

2.2 Proposed two-step coarse search

Now, we are ready to discuss how to calculate the statistics of motion from a valid macroblock sequence located in a shot. The left-hand side of Fig. 2 illustrates a typical shot consisting of *n* anchor frames. For calculating the statistics of motion, the following steps are performed. First, let $\mathbf{m}_{i,j}$ represent the set of motion vectors of a valid macroblock sequence located in the *i*th row, *j*th column of the valid macroblock region. The probability that the quantized (or transformed) motion vectors of this macroblock sequence falls into the bin (*u*,*v*) can be calculated as follows:

$$p(\mathbf{qmv} = (u,v) \mid \mathbf{qmv} \in \mathbf{m}_{i,j}) = \frac{\#\{\mathbf{qmv} \mid \mathbf{qmv} \in \mathbf{m}_{i,j}\}}{N_{\text{MB}}} \tag{3}$$

where **qmv** represents a quantized motion vector obtained from (2) and N_{MB} is the total number of motion vectors in this valid macroblock sequence.

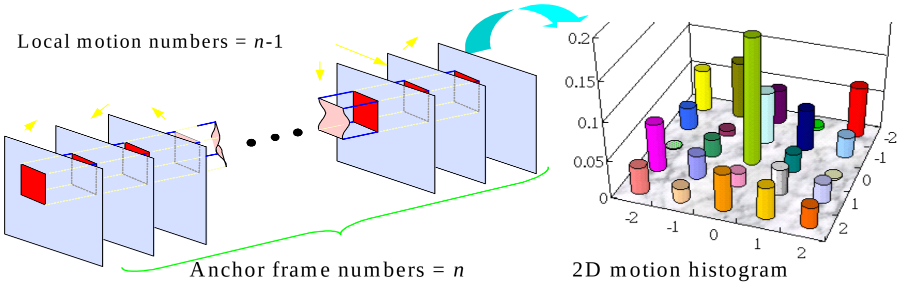


Fig. 2. An example showing how the motion vectors extracted from a macroblock sequence are projected onto the *UV* plane and are calculated as a 2D motion histogram.

At the right-hand side of Fig. 2, we illustrate how to transform *n*-1 motion vectors into the normalized probability distribution map on the *UV* plane. In addition to the probability distribution calculated above, we can also compute the entropy value of every valid macroblock sequence by the following equation:

$$H(S) = -\sum_{u,v} \left\{ p(\mathbf{qmv} = (u,v) \mid \mathbf{qmv} \in \mathbf{m}_{i,j}) \ln p(\mathbf{qmv} = (u,v) \mid \mathbf{qmv} \in \mathbf{m}_{i,j}) \right\} \tag{4}$$

The calculation of the entropy shown in (4) can be used to guide the selection among the constituent shots of an unknown query clip. Usually, a shot having the largest entropy value implies that it contains the most discriminating information that can be used to conduct an efficient search. In this work, a two-step process for efficient video retrieval is proposed. As described above, the complete shot that has the largest entropy value is adopted to execute the first-step coarse-search process, and then, at the second step, we shall adopt a shot that is adjacent to the chosen shot (either preceding or subsequent) and concatenate them together for more accurate search outcome. In other words, the causality is considered.

For comparing two distinct shots, the comparison is of the form of comparing two probability distribution functions. Here, we use the discrete Bhattacharyya distance[16] described as follows to perform the shot comparison task.

$$d(\mathbf{m}_{i,j},\mathbf{m}'_{i,j}) = -\ln \sum_{u,v} \left\{ p(\mathbf{qmv}=(u,v)\,|\,\mathbf{qmv}\in\mathbf{m}_{i,j}) \times p(\mathbf{qmv}'=(u,v)\,|\,\mathbf{qmv}'\in\mathbf{m}'_{i,j}) \right\}^{1/2} \tag{5}$$

where $\mathbf{m}_{i,j}$ and $\mathbf{m}'_{i,j}$ represent the sets of quantized motion vectors extracted, respectively, from the valid macroblock sequence located at the (i,j)th location of two distinct shots.

In order to calculate the overall Bhattacharyya distance between two arbitrary shots, one has to accumulate the measured distances from all macroblock sequence pairs located in the valid macroblock region. The overall similarity, $D(S,S')$, is calculated by

$$D(S,S') = \frac{\sum_{i,j} d(\mathbf{m}_{i,j},\mathbf{m}'_{i,j})}{N} \tag{6}$$

where S and S' represent two distinct shots, and N represents the total number of valid macroblock sequences existing in a shot.

Besides motion, the spatial color information is also an important feature for video retrieval. Color is a highly perceptive image feature and has been extensively used in the literature as a means to achieve content-based image retrieval (CBIR)[17,19]. The advantage of using color histograms is that scale invariance and a high degree of immunity to noise due to motion can be achieved. Because color histograms are also a kind of probability distributions, the discrete Bhattacharyya distance can also be used for measuring the similarity between the color histograms of two shots as follows:

$$d(p,q) = -\ln \sum_i (p_{\mathrm{Y}}(i)\times q_{\mathrm{Y}}(i))^{1/2} - \ln \sum_i (p_{\mathrm{Cb}}(i)\times q_{\mathrm{Cb}}(i))^{1/2} - \ln \sum_i (p_{\mathrm{Cr}}(i)\times q_{\mathrm{Cr}}(i))^{1/2} \tag{7}$$

where p and q are two key-frames of two arbitrary shots, and $p_{\mathrm{Y}}(i)$, $p_{\mathrm{Cb}}(i)$, and $p_{\mathrm{Cr}}(i)$ are the ith bin of the histogram. The subscripts Y, Cb, and Cr denote the three color components of the YCbCr format video.

Because a video has both spatial and temporal dimensions, video retrieval should capture the spatio-temporal contents of video shots to obtain more accurate results. Therefore, we combine the above two similarity metrics of motion and color distributions for shot-level global feature matching to obtain the coarse search results.

3. FINE SEARCH USING LOCAL COLOR HISTOGRAM OF KEY-FRAMES

The above two-step coarse-search can efficiently select s relatively small set of candidate video clips from a large database due to its low computational requirement. The search results, however, may not be accurate enough (say, there may exist many false alarms), since only the coarse global statistics are used to achieve fast video retrieval, which usually cannot capture the spatial layouts and structures of video frames for very accurate retrieval. Using the local spatial layouts and structures can further improve the accuracy of retrieval. Computing and storing the spatial information for every frame of each shot, however, is not efficient in terms of both computational complexity and storage cost. Since the spatial features of adjacent frames are usually similar, using the features of a much smaller set of most representative key-frames of a shot usually can achieve close retrieval performance, while reducing the computational complexity drastically. As a set of key-frames have been chosen for each shot, the problem of video shot matching is now to determine the similarity between the key-frames of two shots. Such problem is similar to CBIR for individual key-frame matching.

3.1 Key-frame selection from a video shot

To reduce the computation of video shot matching while retaining the matching accuracy, the most representative key-frames of the query video shots are extracted for the fine-search step. Fig. 3 illustrates an example of extracting N key-frames from a video shot of M frames, where N = 4 and M = 22 in this example. Let $\mathbf{F} = \{f_1, f_2, \ldots, f_M\}$ be the set of the frames in the video shot, **K** = {k_1, k_2, …, k_N} be a set of key-frames. Then we can partition the shot **F** into a set of N non-overlapping intervals **T** = {T_1, T_2, …, T_N} = {t_0~t_1, t_1~t_2, …, t_{N-1}~t_N} such that the frames in the *i*th interval, $T_i = t_{i-1}$~t_i, are all represented by the *i*th key-frame, k_i. The right-boundary frame of each interval is called a break point. We then define the cost of the *i*th interval represented by the *i*th key-frame as the sum of distance values between the key-frame k_i and the remaining frames in T_i as follows:

$$D(T_i, k_i) = \sum_{t=t_{i-1}}^{t_i} d(t, k_i) \tag{8}$$

where $d(t,k_i)$ is the distance value between frames t and k_i calculated by (7). The goal of the step is two-fold. The first is to determine an appropriate number of key-frames to select from a video shot. After determining the key-frame number for a video shot, the next is to find an optimal combination of **T** and **K** so that the overall cost, $\sum_{i=1}^{N} D(T_i, k_i)$, can be minimized, where the optimum set **K** is called the set of most representative key-frames. This optimization problem, however, usually does not have a closed-form solution, since it is difficult to find a mathematically tractable model to characterize the cost function with a reasonable accuracy. An iterative procedure[13] was proposed to find the optimal key-frame extraction from a video shot. A dynamic programming based scheme[14] was proposed to obtain the optimal key-frame extraction. However, both of the two schemes may be computationally too expensive to be applied in real-time query applications. Another greedy algorithm[15] is proposed for key-frame selection, which has lower complexity, but leads to higher distortion.

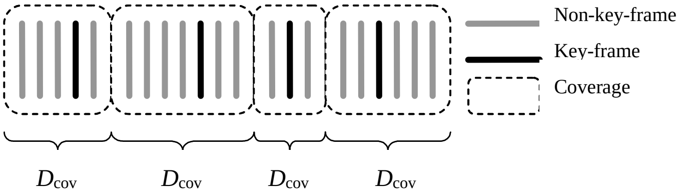


Fig. 3. Illustration of key-frame selection in a video shot.

In this work, we propose a low-complexity non-iterative approach as follows:

Step 1. By summing up the distance values between two adjacent frames for all frames in a shot, we can obtain the overall cumulative distance of video shot as defined below.

$$D_{\text{shot}} = \sum_{m=1}^{M-1} d(f_m, f_{m+1}) \tag{9}$$

The number of key-frames used for the fine-search step is:

$$N = k_{\text{KF}} \frac{D_{\text{shot}}}{M} \tag{10}$$

where k_{KF}, which is an empirically set constant, is to relate the to the key-frame number N to the average cumulative distance of video shot.

Step 2. The average coverage range of each key-frame can be calculated by dividing D_{shot} by N.

$$D_{\text{cov}} = \frac{1}{N} D_{\text{shot}} \tag{11}$$

The video shot is then partitioned into N intervals {t_0~t_1, t_1~t_2, …, t_{N-1}~t_N} each with approximately equal average representation distortion of D_{cov} as follows:

$$\sum_{n=t_{j-1}}^{t_j-1} d(f_n, f_{n+1}) \cong D_{\text{cov}} \quad j = 1, 2, \ldots, N \tag{12}$$

Step 3. For each interval T_i, the key-frame is selected so as to minimizes the distortion $D(T_i,k_i)$:

$$k_i = \arg \min_{t_{i-1} \le k_i \le t_i} D(T_i, k_i), \quad i = 1, 2, ..., N \tag{13}$$

3.2 Final search using local color histograms of key-frames

After extracting the key-frames, we divide the key-frames into 2×2 subimages as shown in Fig. 4. To take into account the local spatial structures to further refine the search results, the color histogram of each subimage is calculated individually.

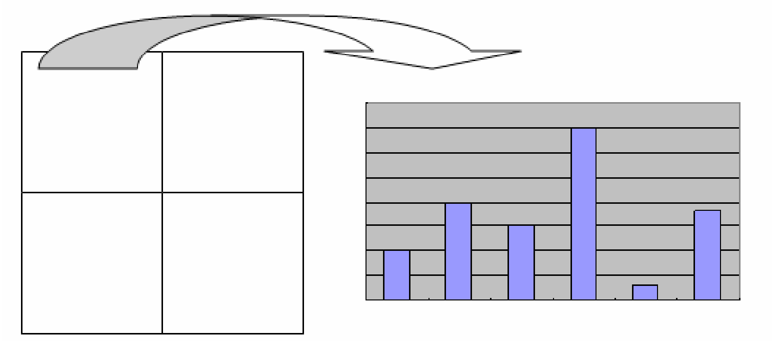


Fig. 4. Each key-frame is divided into 2×2 subimages and the local color histograms of the subimages are calculated individually.

After calculating the local color histograms of each key-frame, we combine the bins of all blocks to form a new distribution and use the color-histogram distance in (7) to measure the similarity between the distributions of two shots to refine the results obtained from the coarse-search steps. By extracting key-frames, significant computation saving can be achieved without sacrificing the retrieval performance. Table 1 compares the retrieval time for the fine-search with and without key-frames from 200 candidate video clips obtained from the coarse-search step if one key-frame is extracted for one shot.

Table 1. Run-time comparison of retrieval with key-frames and with the whole shot

Retrieval / shot-length	with key-frames	with whole shot
236	1 s	8 s
485	1 s	15 s

4. EXPERIMENTAL RESULTS

In order to show the effectiveness of the proposed method, we have tested our algorithm against six digital videos consisting of a total number of 1682 shots. First we used the gradual shot change detection algorithms proposed in [21,22] to extract 1682 complete shots from the six digital videos. The lengths of the six digital videos are 55 minutes (503 shots, documentary, video #1), 52 minutes (405 shots, documentary, video #2), 29 minutes (241 shots, commercial, video #3), 38 minutes (193 shots, news, video #4), 38 minutes (283 shots, sports news, video #5), and 17 minutes (57 shots, home video, video #6), respectively. The reason that we chose these video was due to their variety.

In our experiments, five sample query video shots are selected from the 1682-shot database. For each sample query video clip, the ground-truth of relevant video clips is established by choosing a set of most relevant video shots from the video database by human observations. With the proposed method, each sample query returns a list of matched clips ranked by the similarity order. In the first-step coarse search, a total of 200 candidate shots are retrieved by using the shot-level motion and color statistics from the video database, that is, 100 candidate clips are obtained using the motion feature, and the other 100 clips are obtained using the color feature, respectively. The redundant results obtained from both statistics are then removed. The performance of retrieval is evaluated by the precision and recall rates which are calculated based on the ground-truths as defined in (14) and (15), respectively.

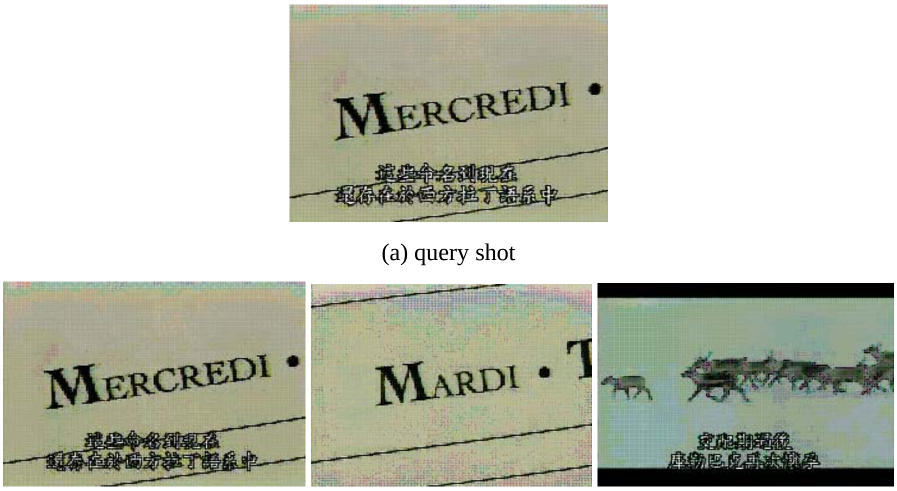
$$\text{Precision} = \frac{N_{\text{hit}}}{N_{\text{hit}} + N_{\text{false}}} \tag{14}$$

$$\text{Recall} = \frac{N_{\text{hit}}}{N_{\text{hit}} + N_{\text{miss}}} \tag{15}$$

where N_{hit} is the number of correct relevant shots returned for the query (i.e., the number of hits), N_{false} is the number of irrelevant shots returned (i.e., the number of false alarms), and N_{miss} is the number of relevant videos that are not returned (i.e., the number of misses).

4.1 Two-step coarse-search results

In the first-step coarse search, a single shot is utilized to perform video retrieval. We extract the global motion and color statistics in the compressed domain from a query shot and then compare these statistics with those in the database. Fig. 5(a) shows a query shot and Fig. 5(b) shows the corresponding top three ranked results out of the 1682-shot database retrieved by the first-step coarse-search.

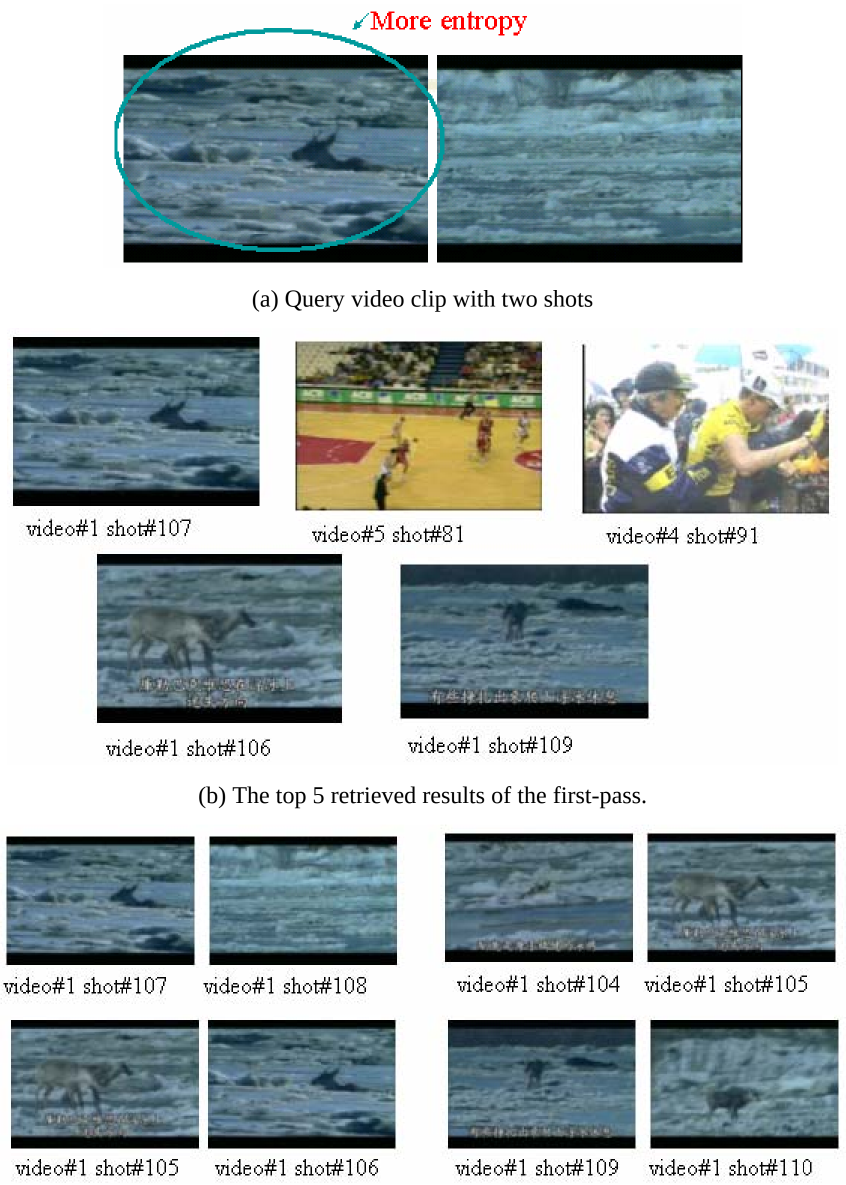


(a) query shot

(b) retrieved shots

Fig. 5. Query example #1: (a) The query shot; (b) The top three retrieved shots out of the 1682-shot database (the left is the top ranked one).

Using the causality between two video shots can effectively improve the coarse-search result. For example, Fig. 6(a) and (b) show that, although the coarse search based on global motion and color statistics is very efficient in terms of computational complexity, the search results may not be very accurate. In the second-step coarse search, we add one more shot (the right hand side of Fig. 6(a)) into the comparison process. Since the causality factor and the global statistics of one more shot are taken into account to enhance the power of the feature set, we can observe that the top four retrieved results (Fig. 6(c)) become much more accurate with respect to the query shot pair.



(a) Query video clip with two shots

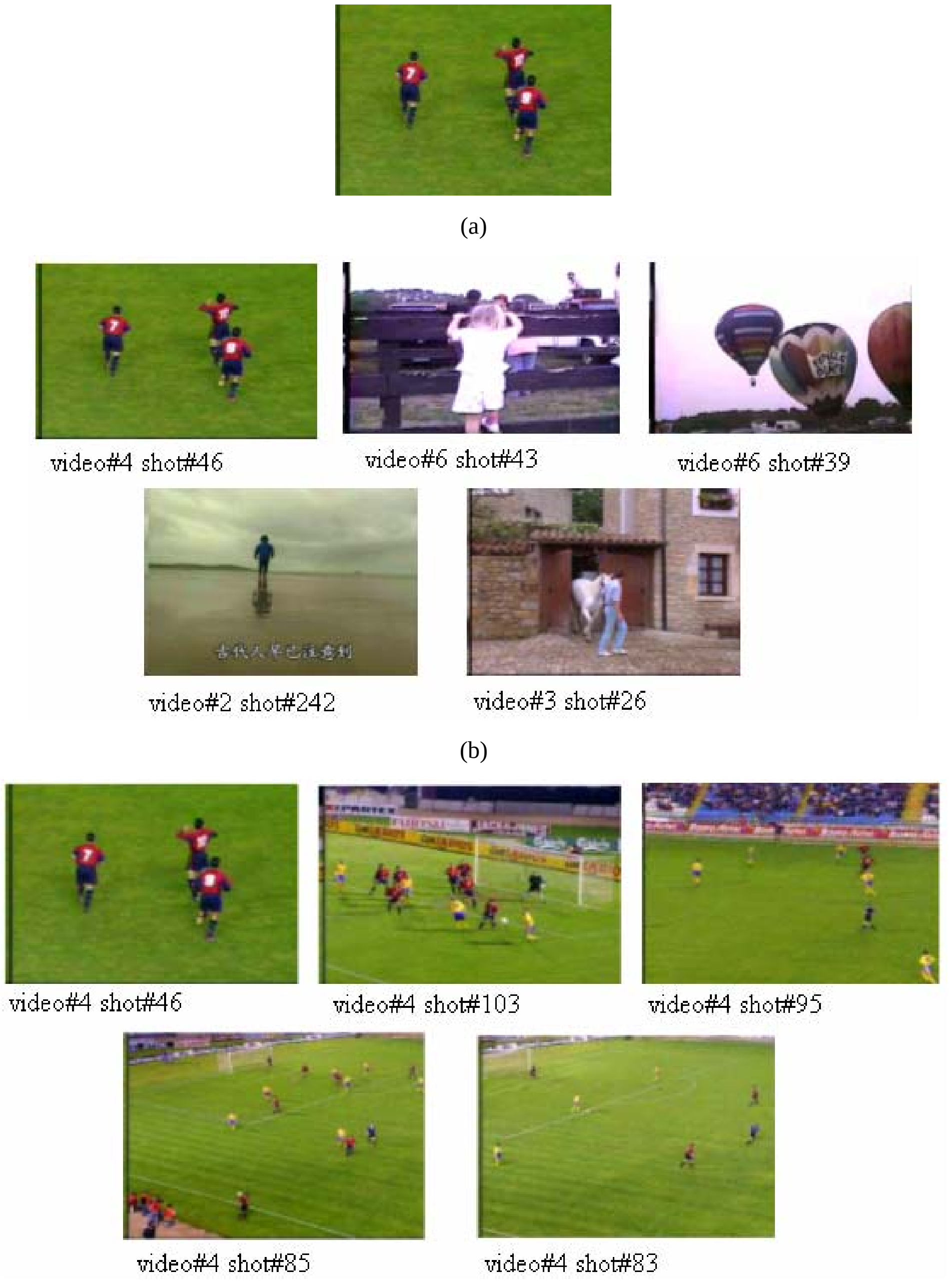
(b) The top 5 retrieved results of the first-pass.

(c) The top 4 retrieved results of the second-pass.

Fig. 6. Query example #2: (a) The query video clip with two shots; (b) The top 5 retrieved shots of the first-pass out of the 1682-shot database; (c) The top 4 retrieved shots of the second-pass out of the candidates from the first-pass.

4.2 Fine-search results

To evaluate the performance of the proposed fine-search scheme, we choose a sport video clip which comprises two shots with multiple motion types for our test. Fig. 7(a) shows the first frame of the query video clip, Fig. 7(b) shows the top five retrieved results of the two-step coarse-search, and Fig. 7(c) shows the top five ranked result of the fine-search using local color histograms of key-frames. Evidently, the combined coarse-to-fine search leads to more accurate results than the two-step coarse search, especially for video clips with multiple motion types. Fig. 8 shows another query example that the proposed coarse-to-fine method can perform much better than a single-pass retrieval.

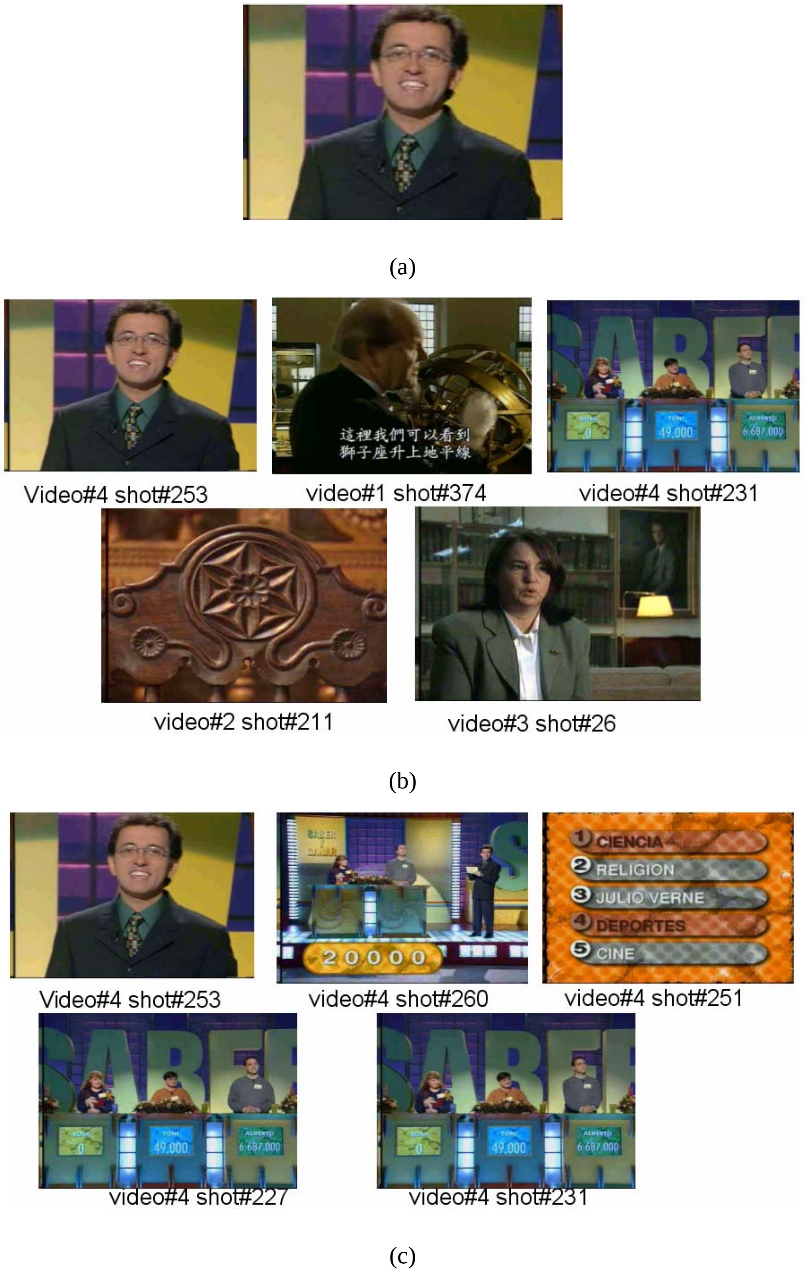


(a)

(b)

(c)

Fig. 7. Query example #3: (a) the query video shot; (b) the top 5 retrieved shots with the two-pass coarse search out of the 1682-shot database; (c) the top 5 retrieved shots with the fine search using the local statistics of key-frames.



(a)

(b)

(c)

Fig. 8. Query example #4: (a) the query video shot; (b) the top 5 retrieved shots with the two-pass coarse search out of the 1682-shot database; (c) the top 5 retrieved shots with the fine search using the local statistics of key-frames.

Fig. 9 shows the average precision-recall curve of the five sample query shots for the first-step coarse-search (using the shot-level motion statistics), the first two-step coarse search (with causality), and the coarse-to-fine search method, respectively. The result shows that the proposed coarse-to-fine search method can achieve good performance with significantly lower computational complexity as compared to single-pass methods.

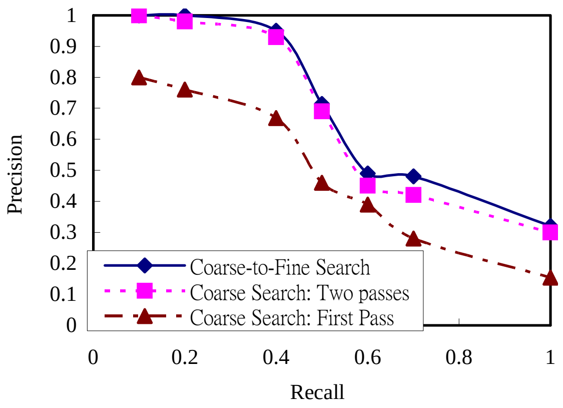


Fig. 9. Average precision-recall curve for the second-pass and our refinement method.

5. CONCLUSION

In this work, we presented a coarse-to-fine fast query-by-clip video retrieval scheme. The proposed scheme involves a two-step coarse search and a fine search. The first-step coarse-search utilizes shot-level global statistics of motion and color of the maximum entropy shot in the query clip as the search features to quickly select a set of candidate video shots from a video database, leading to a much smaller search space than the original database. The causality between two shots is subsequently exploited at the second-step coarse-search to obtain more accurate search outcomes. As a result, the fine-search adopts the local color histograms of key-frames to determine the final best matches from the set of candidate shots obtained from the coarse-search. We have also proposed an efficient scheme for extracting the most representative key-frames from a shot. Experimental results show that our proposed method can provide satisfactory retrieval results with significantly lower complexity than traditional single-pass schemes.

REFERENCES

1. H. S. Chang, S. Sull, and S.-U. Lee, “Efficient video indexing scheme for content-based retrieval,” *IEEE Trans. Circuits Syst. Video Technol.*, no. 9, vol. 8, pp. 1269–1279, Dec. 1999.

2. N. D. Doulamis, A. D. Doulamis, Y. S. Avrithis, and S. D. Kollias, “Video content representation using optimal extraction of frames and scene,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 875–879, Oct. 1998, Chicago, Illinois, USA.

3. R. Fablet and, P. Bouthemy, “Statistical motion based object indexing using optic flow field.’’, *15th IAPR Int. Conf. Pattern Recognition* , 4:3--7, Sept. 2000.

4. Y. F. Ma and, H. J. Zhang, “A new perceived motion based shot content representation.’’, in *Proc. IEEE Int. Conf. Image Processing*, 3:7--10, Oct. 2001.

5. Y. F. Ma and H. J. Zhang, “Motion texture: A new motion based video representation,’’ in *Proc. Int. Conf. Pattern Recognition*, 2:11--15, Aug. 2002.

6. R. L. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic, and E. Persoon, “Visual search in a SMASH system,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 671–674, Sep. 1996, Lausanne, CH.

7. M. M. Yeung and B. Liu, “Efficient matching and clustering of video shots,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 338–341, Oct. 1995. Washington D.C.

8. T. L. Yu and Y. J. Zhang, “Retrieval of video clips using global motion information”, *IEE Electron. Lett. 37(14)*, pp. 893-895, July 2001.

9. S. Lee and M. H. Hayes, “Real-time camera motion classification for content-based video indexing and retrieval using templates,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3664-3667, May 2002.

10. C. Kim and J.-N. Hwang, “Fast and automatic video object segmentation and tracking for content-based applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 122–129, Feb. 2002.

11. S. H. Kim and R. H. Park, “An efficient algorithm for video sequence matching using the modified Hausdorff distance and the directed divergence,” *IEEE Trans. Circuits System Video Technology*, vol. 12, pp. 592-596, July 2002.

12. M. R. Naphade, M. M. Yeung, and B. L. Yeo, “A novel scheme for fast and efficient video sequence matching using compact signatures,” in *Proc. SPIE Conf. Storage and Retrieval for Media Databases*, vol. 3972, pp. 564-572, Jan. 2000.

13. H.-C. Lee and S.-D. Kim, “Iterative key frame selection in the rate-constraint environment,” *Signal Processing*: *Image Commun*., vol. 18, pp. 1-15, 2003.

14. X. S. Zhou and S.-P. Liou, “Optimal nonlinear sampling for video streaming at low bit rates,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 535-545, June 2002.

15. Z. Li, A. Katsaggelos, and B. Gandhi, “Temporal rate-distortion optimal video summary generation,” in *Proc. IEEE Conf. Multimedia & Expo*, pp. 693-696, July 2003, Baltimore, MD.

16. L.-F. Chen, H.-Y. M. Liao, J.-C. Lin, and C.-C. Han, “Why recognition in a statistics-based face recognition system should be based on the pure face portion: a probabilistic decision-based proof,” *Pattern Recognition*, vol. 34, no. 5, pp. 1393-1403, 2001.

17. A. D. Bimbo, *Visual Information Retrieval*, Morgan Kaufmann, 1999.

18. G. Lu, *Multimedia Database Management Systems*, Artech House, 1999.

19. H. Lu, B. C. Ooi, and K. L. Tan, “Efficient image retrieval by color contents,” in *Proc. Int. Conf. Application of Database*, pp. 95-108, 1994.

20. M. Swain and D. Ballard, “Color indexing,” *Int. J. Computer Vision*, 7(1), pp. 11-32, 1991.

21. C.-C. Shih, H.-Y. M. Liao, and H.-R. Tyan, “Shot change detection based on the Reynolds transport theorem,” in *Proc. Second IEEE Pacific Rim Conf. Multimedia*, Oct. 2001, Beijing, China.

22. C. W. Su, H. Y. M. Liao, H. R. Tyan and, L. H. Chen, “A motion-tolerant dissolve detection algorithm,’’ accepted and to appear *IEEE Transactions on Multimedia.*

23. G. Davenport, T.A. Smith and, N. Pincever, “Cinematic primitives for multimedia,’’ *IEEE Computers Graphics & Applications*, 11(4):67--74, July 1991.

24. M. V. Srinivasan, S. Venkatesh and, R. Hosie, “Qualitative estimation of camera motion parameters form video sequences,’’ *Pattern Recognition*, vol. 30, no. 4, pp. 593--606, 1997.

25. A. K. Jain, A. Vailaya and, W. Xiong, “Query by video clip,’’, *Multimedia System*, vol. 7, pp. 369--384, 1999.