10-2007

OM-based video shot retrieval by one-to-one matching

Yuxin PENG

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Jianguo XIAO

Multimed Tools Appl (2007) 34:249–266
DOI 10.1007/s11042-006-0085-4

OM-based video shot retrieval by one-to-one matching

Yuxin Peng · Chong-Wah Ngo · Jianguo Xiao

Published online: 16 February 2007

Abstract This paper proposes a new approach for shot-based retrieval by optimal matching (OM), which provides an effective mechanism for the similarity measure and ranking of shots by one-to-one matching. In the proposed approach, a weighted bipartite graph is constructed to model the color similarity between two shots. Then OM based on Kuhn–Munkres algorithm is employed to compute the maximum weight of a constructed bipartite graph as the shot similarity value by one-to-one matching among frames. To improve the speed efficiency of OM, two improved algorithms are also proposed: bipartite graph construction based on subshots and bipartite graph construction based on the same number of keyframes. Besides color similarity, motion feature is also employed for shot similarity measure. A motion histogram is constructed for each shot, the motion similarity between two shots is then measured by the intersection of their motion histograms. Finally, the shot similarity is based on the linear combination of color and motion similarity. Experimental results indicate that the proposed approach achieves better performance than other methods in terms of ranking and retrieval capability.

Keywords Shot-based retrieval · OM · Color and motion similarity

1 Introduction

Due to the drastic advances in multimedia and internet applications, the effective yet efficient techniques for video retrieval are increasingly demanded. One critical component

Y. Peng (✉) · J. Xiao
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
e-mail: pengyuxin@icst.pku.edu.cn

J. Xiao
e-mail: xiaojianguo@icst.pku.edu.cn

C.-W. Ngo
Department of Computer Science, City University of Hong Kong, Kowloon, Hongkong, China
e-mail: cwngo@cs.cityu.edu.hk

in these techniques is the similarity measure of video content. In general, the techniques in video retrieval can be grouped into two categories: shot-based retrieval and clip-based retrieval. A shot is a series of frames with continuous camera motion, while a clip is a series of shots that are coherent from the narrative as well as the user's point of view. Shot-based retrieval, as the basis for video retrieval, clustering and summarization, remains a challenging problem and has attracted numerous research attentions. For instance, in TRECVID, shot-based retrieval is addressed over the past few years [12, 15, 16].

Related works in shot-based retrieval include [1, 7, 8, 11, 20, 23]. In [8, 20], one keyframe is extracted to represent the content of a shot using unsupervised clustering method. In [23], nearest feature line (NFL) is employed to extract the keyframes. After keyframes are extracted, shot-based similarity measure is equivalent to image-based similarity measure. As a result, shot-based retrieval can be tackled in a similar way as image retrieval. However, in addition to image information, video also contains spatio-temporal and motion information. The approaches in [8, 20, 23] did not exploit the special information existing in videos. In [7, 11], subshot is proposed for shot-based similarity measure. A shot with significant content changes is represented by several coherent subshots. The shot similarity is measured based on their corresponding subshots. In [11], subshots are segmented based on its motion content, and keyframes are extracted and constructed to represent subshots of different motion content. For example, a static subshot is represented by one keyframe, a pan subshot is represented by constructing a panoramic image, and a zoom subshot is represented by two selected keyframes before and after zoom. The shot similarity is equal to the average of maximum similarity and the second largest similarity value in all pair of keyframes. In [7], dominant color histograms (DCH) and spatial structure histograms (SSH) are proposed to extract and represent subshot, the similarity between two shots is equal to the maximum similarity of their subshots. The methods in [7, 11] exploit the motion and spatio-temporal information existing in shots, however, the methods using the maximum and the second largest similarity value cannot fully and objectively measure the shot similarity. The method in [1] assumes the frames in two shots are similar in temporal order, dynamic programming is employed to measure the shot similarity. But the assumption is not always correct. Besides, the retrieval speed is slow because the similarity measure is based on every pairs of frames between two shots. The method in [4] proposes to extract one keyframe within each shot as a representative image. Image features extracted from the reprehensive images are then used for retrieval. The method in [4] employs three different types of image features: color histogram, textures and edges. The image is split into a 5-by-5 grid that captures some spatial locality of information. In [6], a multi-level matching scheme is proposed to recursively measure sequence similarity at shot-scene-video level. At the shot level, two types of representation (sequence and set) are proposed for matching. Recently, a region-based shot retrieval system is proposed and latent semantic analysis (LSA) is employed to model the visual content of shot sequence for object retrieval in [17].

The above methods [1, 4, 6–8, 11, 17, 20, 23] mainly employ color feature for shot similarity measure. Recently, the motion feature and similarity are also proposed for shot similarity measure [2, 3, 19]. In [2], color, texture and motion features are used for shot-based retrieval, these features are represented by color histogram, Gabor texture features and motion histogram. The shot similarity is measured by the linear combination of these features similarity. In [3], *ClassView* is proposed for hierarchical video shot classification, indexing, and accessing. A nine-dimensional directional motion histogram and other visual features are extracted to represent a video shot. In [19], *ViBE* is proposed to index and

browse videos. Motion histogram derived from the motion vectors of macro blocks is constructed to quantify the motion content of a shot.

Besides shot-based retrieval, clip-based retrieval has also been studied [5, 13, 22]. Clip-based retrieval, in general, is built upon the shot-based retrieval. Besides relying on the visual similarity between shots, clip-based retrieval should consider the inter-relationship among video shots. In [13], maximum matching (MM) and optimal matching (OM) are proposed to measure clip-based similarity. MM is able to rapidly filter irrelevant video clips, while OM is capable of ranking the clip similarity according to visual and granularity factors. Temporal order and interference factors are also measured based on the output of OM. In [5], temporal order is imposed as a hard constraint. In other words, similar clips must obey the same temporal order. As a result, video clips with similar content but different shot order will not be retrieved. Recently, an index structure based on multi-resolution KD-tree is proposed in [22] to further speed up clip-based retrieval.

In this paper, we propose new algorithms to extend OM to shot-based retrieval. OM provides an effective mechanism for shot similarity measure and ranking by one-to-one matching. The major contributions of the proposed approach are as follows:

- *Color Similarity measure by OM*. A graph matching algorithm, namely optimal matching (OM) [13], is adopted for color similarity measure between two shots. A weighted bipartite graph is constructed to model the similarity between two shots: every vertex in a bipartite graph represents one keyframe in a shot, and the weight of an edge represents the color similarity for a pair of keyframes between two shots. Then OM based on Kuhn–Munkres algorithm is employed to compute the maximum weight of a constructed bipartite graph as the similarity value between two shots by guaranteeing the one-to-one matching among frames.
- *Two improved approaches for OM*. To improve the speed efficiency of OM, two improved approaches are proposed as follows: bipartite graph construction based on subshots and bipartite graph construction based on the same number of keyframes in shots.

To effectively measure shot similarity, the color and motion similarity between two shots are jointly measured. In motion similarity measure, similar to [2], a motion histogram is constructed to represent a shot, the motion similarity is measured by the intersection between two motion histogram. The final similarity is based on the linear combination of color and motion similarity.

The rest of this paper is organized as follows. Section 2 is the highlight of this paper, which describes the color similarity measure by OM. Section 3 presents the motion similarity measure by motion histogram. Based on the results of color and motion similarity measure, Section 4 describes the similarity measure between two shots. Section 5 shows the experimental results while Section 6 concludes this paper.

2 Color similarity measure

In color similarity measure, the weighted bipartite graph of two shots is constructed as follows:

- Let $X=\{x_1, x_2,..., x_p\}$ as a query shot with p frames, and x_i represents a frame in X.
- Let $Y_k=\{y_1, y_2,..., y_q\}$ as a shot in the video database with q frames, and y_j is a frame in Y_k.

- Let $G_k=\{X, Y_k,E_k\}$ as a weighted bipartite graph, where $V_k = X \cup Y_k$ is the vertex set, $E_k=\{\omega_{ij}\}$ is the edge set, and ω_{ij} represents the color similarity value between x_i and y_j.

In the proposed approach, the color similarity value ω_{ij} is computed by the histogram intersection [18] as the follows:

$$\omega_{ij} = \frac{1}{A(x_i, y_j)} \sum_h \sum_s \sum_v \min\left\{H_i(h,s,v), H_j(h,s,v)\right\} \tag{1}$$

$$A(x_i, y_j) = \min\left\{\sum_h \sum_s \sum_v H_i(h,s,v), \sum_h \sum_s \sum_v H_j(h,s,v)\right\} \tag{2}$$

We use 3D HSV color histogram for shot similarity measure. According to human perception, hue is more effective than saturation and intensity in color similarity measure. In our approach, hue is quantized into 18 bins while saturation and intensity are quantized into 3 bins, respectively. The quantization provides 162 ($18\times3\times3$) distinct color sets. After $G_k=\{X, Y_k,E_k\}$ is constructed, OM based on Kuhn–Munkres algorithm [21] is employed to measure similarity between X and Y_k, the algorithm is given in Fig. 1.

The computational complexity of Kuhn–Munkres algorithm is $O(n^4)$, where $n = p + q$, is the total number of vertex in G_k. The color similarity $Similarity_{color}(X, Y_k)$ between two shots X and Y_k is defined as follows:

$$\text{Similarity}_{\text{color}}(X, Y_k) = \frac{\omega_{OM}(X, Y_k)}{\min(p, q)} \tag{3}$$

where $\omega_{OM}(X,Y_k)$ is the total weight after OM.

Fig. 1 Kuhn–Munkres algorithm for OM

1. Start with the initial label of $l(x_i) = \max_j(\omega_{ij})$ and $l(y_j) = 0$, where $i = 1,2,..., p$ and $j = 1,2,..., q$.
2. Compute $E_l = \left\{(x_i, y_j) \middle| l(x_i) + l(y_j) = \omega_{ij}\right\}, G_l = (X, Y_k, E_l)$ and one matching M in G_l.
3. If M contains all the vertices in X, M is the optimal matching of G_k and the algorithm ends. Otherwise, go to step 4.
4. Find a vertex $x_i \in X$ that is not inside M. Set $A \leftarrow \{x_i\}$ and $B \leftarrow \phi$.
5. Let $N_{G_l}(A) \subseteq Y_k$ be the set of vertices that matches the vertices in set A. If $N_{G_l}(A) = B$, then go to step 9, otherwise go to step 6.
6. Randomly find a vertex $y_j \in N_{G_l}(A) - B$.
7. If there exists a node $z \in X$ such that $(z, y_j) \in M$, set $A \leftarrow A \cup \{z\}, B \leftarrow B \cup \{y_j\}$ and go to step 5. Otherwise, go to step 8.
8. There exists an augmenting path P from x_i to y_j. Set $M \leftarrow M \oplus E(P)$ and go to step 3.
9. Compute $a = \min_{\substack{x_i \in A \\ y_j \notin N_{G_l}(A)}} \left\{l(x_i) + l(y_j) - \omega_{ij}\right\}$, then construct a new label $l'(v)$ by
$$l'(v) = \begin{cases} l(v) - a & v \in A \\ l(v) + a & v \in B \\ l(v) & otherwise \end{cases}$$
Compute $E_{l'}, G_{l'}$ based on l'.
10. Set $l \leftarrow l', G_l \leftarrow G_{l'}$, go to step 5.

The above approach can measure effectively the shot similarity. However, some shots often have thousands of frames, it is time consuming for Kuhn–Munkres algorithm to compute a bipartite graph composed of hundreds of vertices. In addition, considering the content redundancy in a shot, for example, a static shot may include thousands of frames, one frame is indeed enough to be selected to represent the shot content. To speed up the matching time, the two improved approaches are proposed as follows.

(1) *Bipartite graph construction based on subshots*.

A shot with significant content changes is represented by several coherent subshots. The method in [11] is utilized to segment a shot into several coherent subshots based on its motion content. Then keyframes are extracted and constructed to represent subshots of different motion content. The detail is presented in Table 1.

According to the method in Table 1, the matching time of Kuhn–Munkres algorithm can be speed up significantly. However, some shots only have one subshot and one keyframe. For example, one keyframe in the static shot, and one panoramic keyframe in the pan shot. In this situation, Kuhn–Munkres algorithm is employed to compute the maximum similarity value with one keyframe. To solve this problem, the following approach is employed.

(2) *Bipartite graph construction based on the same number of keyframes in shots.*

In the proposed approach, the complete bipartite graph can be constructed based on the same number of keyframe. In this way, the problem in method (1) can be solved. The shot similarity can be efficiently measured by one-to-one matching among keyframes. In the constructed bipartite graph, our idea is not to constrain the number of keyframes, but to restrict such that the same number of keyframes from both shots is used for comparison. In general, the larger the number of keyframe, the slower the retrieval speed. In the experiment, we assign the number of keyframe to be three, which leveraging the retrieval performance and speed. Although keyframe used by method (2) may contain redundant information, the final similarity measure will not be seriously affected.

3 Motion similarity measure

In shot-based retrieval, the global motion features can be fully employed for shot similarity measure. For example, in sport video, the same sport classes often exhibit similar motion patterns. For instance, in diving videos, the vertical up-and-down due to camera tilting is a common motion feature. In other video genres, different editions of the same shot also share similar motion feature.

Similar to [2], in this paper, motion histogram of shot is constructed based on motion vector field (MVF). For a given frame in a video, MVFs are extracted between the current and the next frame, and the motion characteristics are calculated. In our approach, MVFs

Table 1 Subshot selection and construction in [11]

Subshot	Keyframe
Static	Select one frame
Pan or tilt	Form a new panoramic image
Zoom	Select first and last frames
Multiple motion	Reconstruct background
Indeterministic	Select one frame

are extracted from MPEG video directly without decompression. Motion histogram of a shot is calculated based on two inductors: *angle inductor* and *intensity inductor*. The *angle inductor* induces the direction of motion vector, while *intensity inductor* induces motion energy or activity. They are calculated as follows:

$$\text{angle}(i,j) = arctg\left(\frac{dy_{i,j}}{dx_{i,j}}\right) \tag{4}$$

$$\text{intensity}(i,j) = \sqrt{dx_{i,j}^2 + dy_{i,j}^2} \tag{5}$$

where ($dx_{i,j}$, $dy_{i,j}$) denote two components of motion vector. The angle in 2π is quantized into t angle ranges. Then intensity in each angle range is accumulated over a shot to form a motion histogram with t bins, denoted by $H_X(angle)$, where X is the shot, $angle \in [1, t]$. In this implementation, t is set to 8. In addition, only the MVFs in P-frame are considered in order to reduce computational complexity. Finally, the motion similarity between two shots X and Y_k is defined as follows:

$$\text{Similarity}_{\text{motion}}(X, Y_k) = \frac{1}{A(H_X, H_{Y_k})} \sum_{angle} \min\{H_X(\text{angle}), H_{Y_k}(\text{angle})\} \tag{6}$$

$$A(H_X, H_{Y_k}) = \max\left\{\sum_{angle} H_X(\text{angle}), \sum_{angle} H_{Y_k}(\text{angle})\right\} \tag{7}$$

Note that the motion similarity of shots is normalized by $\max\left\{\sum_{angle} H_X(\text{angle}), \sum_{angle} H_{Y_k}(\text{angle})\right\}$ not $\min\left\{\sum_{angle} H_X(\text{angle}), \sum_{angle} H_{Y_k}(\text{angle})\right\}$ as the color similarity in (3). For color similarity measure, when the size of keyframes is different, (e.g., one of them is mosaic keyframe), we use min so as to emphasize the degree of intersection, and not to degrade its similarity by the scenes or objects in the larger keyframe not found in the smaller keyframe. When the size of keyframes is same, min is simply equal to the size of two compared keyframes. For motion similarity measure, note that motion histogram is constructed by computing the motion vector field in P-frames. If min is used, the similarity is bias towards shots with few motion-coded blocks. Take the motion histograms of two irrelevant shots as an example, the min term, which serves as a normalization factor, will magnify the final similarity, if only few motion-coded blocks are found in one of the shots.

4 Shot similarity measure

Based on the results of color and motion similarity measure, the similarity measure between two shots X and Y_k is defined as follows:

$$\text{Similarity}(X, Y_k) = \omega_1 \text{Similarity}_{\text{color}}(X, Y_k) + \omega_2 \text{Similarity}_{\text{motion}}(X, Y_k) \tag{8}$$

where ω_1 and ω_2 are the weights of color and motion similarity, respectively, and $\omega_1 + \omega_2 = 1$.

The value of ω_i controls the ranking of similar shots. According to human perception, color similarity, in general, is more effective than motion similarity in most video retrieval related tasks. As shown in Tables 4, 5 and Fig. 2 of Section 5, all methods using color similarity (methods II–

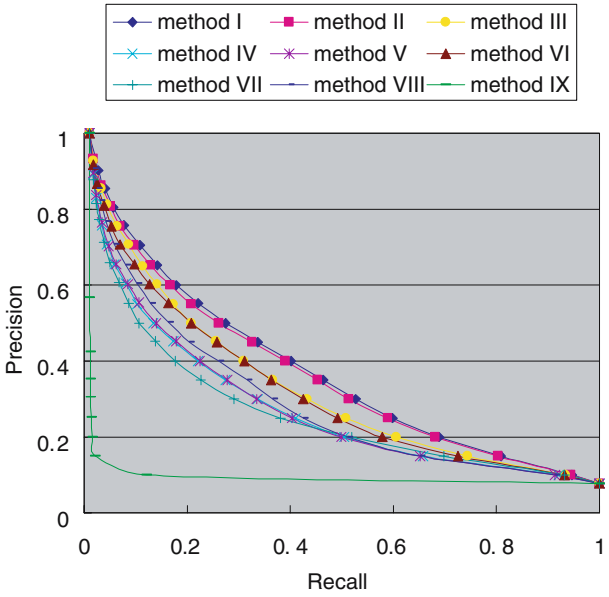

Fig. 2 Precision-recall curves for nine different approaches

VIII) achieve better results than that using motion similarity (method IX) in terms of evaluation based on AR & ANMRR and precision & recall. Thus, we set $\omega_1 > \omega_2 (\omega_1 = 0.7, \omega_2 = 0.3)$ in the experiments. These values can also be set based on user preference.

5 Experimental results

To evaluate the performance of the proposed approach, we set up a database that consists of 3 h videos. The genres of videos include sports, movies and commercials collected from TV stations.

Table 2 Query classes and the number of their relevant shots

Query class	Number of relevant shots
1. Gym	117
2. Bicycle	14
3. Swimming	89
4. Judo	80
5. Weight lifting	36
6. Volleyball	484
7. Football	287
8. Basketball	19
9. Field Hockey	67
10. Fence-play	49
11. An office setting	15
12. Doctors salvaging patient	36
13. Policeman	51
14. A room setting, i.e. Some people gambling	49
15. One person sculpturing	16
Total	1409

We employ the method in [10] to detect the shot boundary of videos in database. In total, there are 3,392 shots. Most of the shots are correctly detected except with few missing shot boundaries. We select 15 classes of queries for experiment, as shown in Table 2. The relevant shots of 15 query classes are manually identified as ground truth set. For the query classes related to sport (e.g., swimming, basketball), the shots with the same sport class are regarded as the relevant shots. For other classes of queries, we browse through the videos and identify the relevant shots manually. In the experiments, we use all the relevant shots of query classes for testing. As shown in Table 2, totally there are 1,409 queries of 15 classes being tested. For performance comparison, nine approaches are evaluated. Table 3 gives a brief description of these approaches.

5.1 Shot ranking

AR (average recall) and ANMRR (average normalized modified retrieval rank) are adopted to evaluate the performance of shot ranking [9]. The values of AR and ANMRR range from [0, 1]. A *high* value of AR denotes the superior ability in retrieving relevant shots, while a *low* value of ANMRR indicates the high retrieval rate with relevant shots ranked at the top (see Appendix for details).

Table 3 Descriptions of nine methods in experimental comparison

Feature	Method		Description
	I		Linear combination of color similarity and motion similarity. The first, middle and last frames in every shot are extracted as keyframes to construct the complete bipartite graph. The weight of color is 0.7, and the weight of motion is 0.3.
OM	II	162 bins in HSV color space are used to represent the color features, and the color similarity is measured by the histogram intersection.	Color similarity measure by OM based on three keyframes in every shot. The first, middle and last frames in every shot are extracted as keyframes to construct the complete bipartite graph.
	III		Color similarity measure by OM based on subshot representation.
[11]	IV		Motion-based shot representation and similarity measure.
One frame	V		Extracting one keyframe within each shot as a representative image.
Variants of methods based on [4]	VI	One keyframe is extracted within each shot. Keyframes are split into 5-by-5 grids. Each grid is presented by its color histogram in 125 dimensions. Each color channel is represented in 5 dimensions and plotted in a 3D histogram. For each image, the dimension of the color histogram is 3125 (5×5×125).	HSV color space
	VII		HVC color space
	VIII		RGB color space
Motion Similarity	IX	A motion histogram is constructed to represent every shot, the motion similarity between two shots is measured by the intersection between their motion histograms.	

Table 4 AR for performance comparison of nine methods

Queries	OM			IV	V	Methods based on [4]			IX
	I	II	III			VI	VII	VIII	
1	0.5873	0.5773	0.6171	0.4484	0.4409	0.5026	0.5512	0.5413	0.1560
2	1.0000	1.0000	1.0000	0.9375	0.9375	1.0000	1.0000	1.0000	0.2500
3	0.5872	0.5863	0.5839	0.5537	0.5618	0.5883	0.4780	0.5815	0.1745
4	0.4375	0.4283	0.4161	0.3738	0.3969	0.3916	0.3380	0.3397	0.1689
5	0.7731	0.7747	0.6458	0.5054	0.4552	0.6127	0.4144	0.5340	0.0949
6	0.6767	0.6737	0.6120	0.5522	0.5321	0.5936	0.5816	0.5342	0.3075
7	0.6536	0.6510	0.6075	0.5957	0.6027	0.6157	0.6109	0.6173	0.3492
8	0.7064	0.7036	0.7867	0.7784	0.4986	0.7036	0.3934	0.7590	0.0886
9	0.7135	0.7155	0.6086	0.6026	0.6407	0.7124	0.4937	0.6137	0.0920
10	0.7530	0.7505	0.7430	0.7318	0.7876	0.8051	0.5881	0.4019	0.1395
11	0.4800	0.4800	0.6400	0.5200	0.3600	0.4800	0.6400	0.6000	0.2400
12	0.4715	0.4807	0.4560	0.4267	0.4321	0.5424	0.4815	0.4267	0.0810
13	0.4168	0.3910	0.3829	0.2080	0.2860	0.2826	0.3560	0.3752	0.1346
14	0.4107	0.3644	0.2262	0.2232	0.2711	0.2882	0.1824	0.2137	0.0925
15	0.5278	0.5278	0.3889	0.3889	0.3611	0.4722	0.2778	0.3611	0.2222
Average	0.6130	0.6070	0.5810	0.5231	0.5043	0.5727	0.4925	0.5266	0.1728

Experimental results on AR and ANMRR for nine methods are shown in Tables 4 and 5. The findings of experimental results are summarized as follows:

- Our three methods using OM (methods I, II and III) outperform other methods (methods IV, V, VI, VII and VIII) in terms of AR and ANMRR. The main reasons are: OM provides an effective mechanism for shot similarity measure and ranking by one-to-one matching. In several query classes, the three methods are not always better than other methods. But for most of the query classes, the proposed approaches are always better than other methods. The same conclusion can also be drawn from the global precision-recall curve in Section 5.2.
- Method I achieves the best AR and ANMRR among the nine methods. Comparing method I with method II, method I adds the motion features based on method II. The result indicates motion features is useful for shot-based similarity measure.
- Methods II and III only utilize color features. In method III, some shots only include one subshot and one keyframe based on camera motion, and then OM is employed to compute the maximum similarity value with one keyframe. While in method II, three keyframes are extracted in every shot, then a complete bipartite graph is constructed, shot similarity can be efficiently measured by OM. The problem in method III can be solved in method II, so the method II outperforms the method III in terms of AR and ANMRR.
- Comparing with two methods using subshot (methods III and IV), method III achieves better AR and ANMRR than method IV. Because both methods utilize the same keyframes and feature for every shot, the result also indicates OM is effective for shot similarity measure.
- All methods using color similarity (II–VIII) achieve better results than that using motion similarity (IX). This result indicates that motion feature with histogram representation, although is useful for shot similarity measure, is not as effective as the color feature with histogram representation. So, in shot similarity measure, the degree of color similarity should carry more weight than that of motion similarity.

Table 5 ANMRR for performance comparison of nine methods

Queries	OM			IV	V	Methods based on [4]			IX
	I	II	III			VI	VII	VIII	
1	0.5311	0.5418	0.5133	0.6567	0.6784	0.6089	0.5677	0.5712	0.9074
2	0.0043	0.0043	0.0000	0.1681	0.0905	0.0086	0.0302	0.0000	0.7241
3	0.5150	0.5164	0.4949	0.5650	0.5689	0.5226	0.6145	0.4726	0.8826
4	0.6438	0.6510	0.6758	0.7090	0.6842	0.6765	0.7450	0.7274	0.8905
5	0.2929	0.2931	0.4371	0.5639	0.6183	0.4667	0.6384	0.5447	0.9177
6	0.4203	0.4247	0.4887	0.5523	0.5656	0.5121	0.5287	0.5498	0.7861
7	0.4596	0.4637	0.5127	0.5226	0.5222	0.4868	0.5556	0.5141	0.7748
8	0.4429	0.4613	0.3633	0.4041	0.5872	0.4320	0.6744	0.4010	0.9166
9	0.3993	0.3948	0.4695	0.5034	0.4768	0.4058	0.5770	0.4842	0.9359
10	0.3571	0.3573	0.3986	0.4119	0.3326	0.3117	0.5239	0.6768	0.9069
11	0.5503	0.5568	0.3903	0.5481	0.6346	0.5784	0.4270	0.4616	0.7708
12	0.6274	0.6183	0.6546	0.6770	0.6682	0.5713	0.6610	0.6834	0.9429
13	0.6282	0.6566	0.6843	0.8283	0.7675	0.7543	0.7064	0.6779	0.9051
14	0.6688	0.7128	0.8191	0.8356	0.8034	0.7954	0.8492	0.8248	0.9335
15	0.5155	0.5039	0.6085	0.6318	0.6357	0.5556	0.7248	0.6473	0.7597
Average	0.4704	0.4771	0.5007	0.5719	0.5756	0.5124	0.5883	0.5491	0.8636

Figures 3, 4, 5 and 6 show the experimental results of method I for some query classes. The shot in upper left corner is the query shot, and the relevant shots are ranked from left to right and from upper to bottom.

5.2 Shot retrieval

Precision and recall are adopted to evaluate shot retrieval performance. The precision and recall are defined as follows:

$$\text{Precision} = \frac{\text{Number of relevant shots being retrieved}}{\text{Number of shots being retrieved}} \tag{9}$$

$$\text{Recall} = \frac{\text{Number of relevant shots being retrieved}}{\text{Number of relevant shots}} \tag{10}$$

The precision-recall curve for nine methods is shown in Fig. 2. *X*-axis represents recall, and *Y*-axis represents precision. In Fig. 2, the result of precision-recall curve in upper right corner is always better than that in bottom left corner. For example, the performance of method I is the best, while that of method IX is relatively worst. The conclusions are similar with Section 5.1 and briefly described as follows:

- Our three methods using OM (methods I, II and III) outperform other methods (methods IV, V, VI, VII and VIII) in terms of precision and recall. The result indicates OM is effective for shot-based retrieval.
- Although method I achieves the best precision and recall in the nine methods, its performance is close to method II. The result indicates: on one hand, motion

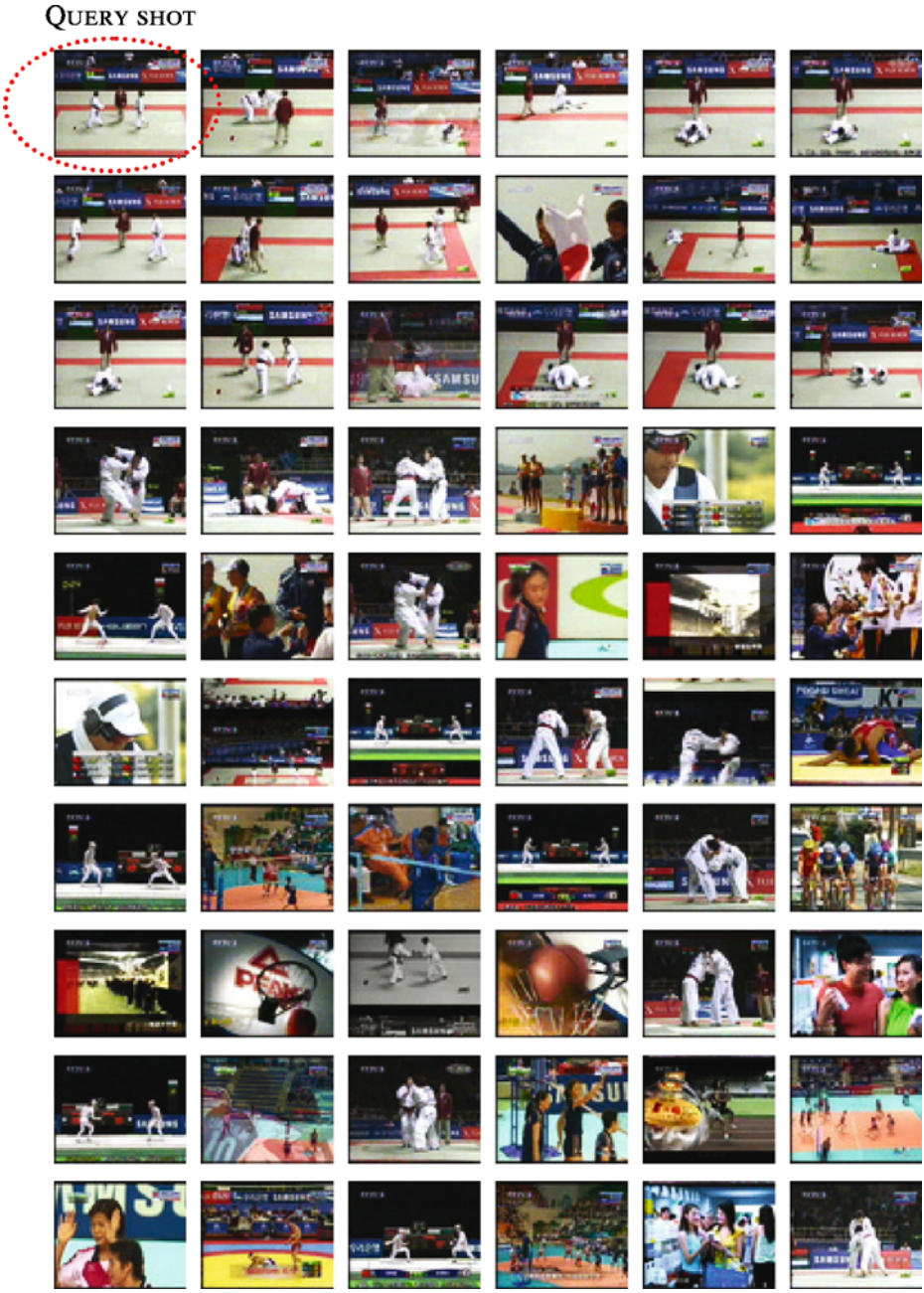

Fig. 3 Query results for a judo shot

feature is useful for shot-based similarity measure, on the other hand, as presented in [4], motion feature is also noisy, which is not as effective as color feature.

- In two methods using color features (methods II and III), method II outperforms method III in terms of precision and recall. The main reasons are: In method III,

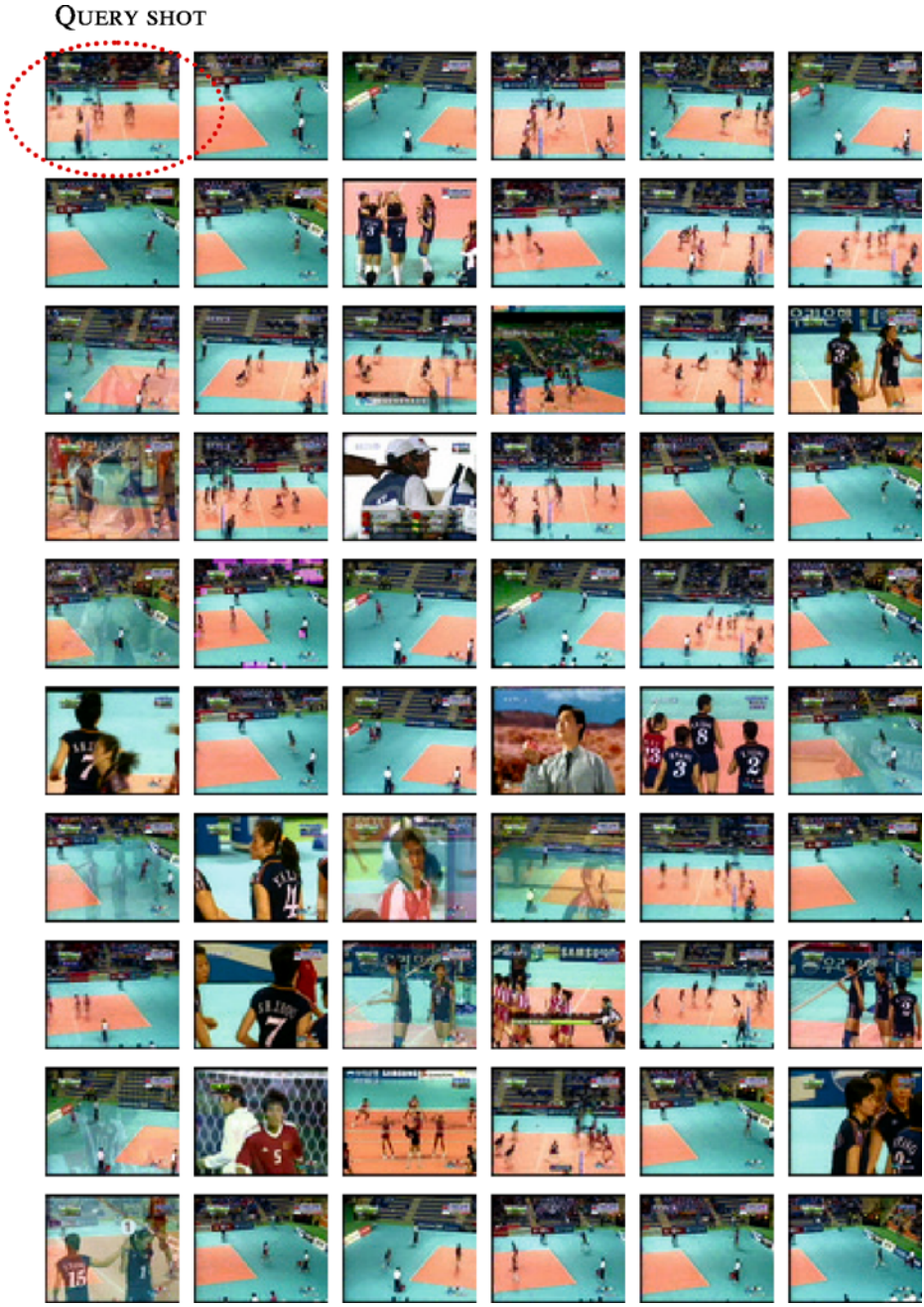

Fig. 4 Query results for a volleyball shot

some shots only include one subshot and one keyframe based on camera motion, and then OM is employed to compute the maximum similarity with one keyframe. While in method II, three keyframes are extracted in every shot for bipartite graph construction, shot similarity can be efficiently measured by OM.

Fig. 5 Query results for a football shot

- Methods III and VI have similar results, although method III achieves better performance than method VI in terms of precision and recall. The result also indicates, as aforementioned, some shots only include one subshot and one keyframe in method III, which is not so effective for shot similarity measure by OM.

Fig. 6 Query results for a fence-play shot

- Comparing with two methods using subshot (methods III and IV), method III is better than method IV in terms of precision and recall. Because both methods utilize same keyframes and feature for every shot, the result also indicates that OM is effective for shot similarity measure. The main reasons are: OM provides a good mechanism for shot similarity measure by one-to-one matching.
- Similar to the performance based on AR and ANMRR in Section 5.1, all methods of color similarity (methods II-VIII) are better than the method of motion similarity (method IX) in terms of precision and recall.

5.3 Retrieval speed

The average retrieval time (ART) of queries is shown in Table 6 for the nine experimented approaches. The testing is conducted on a Pentium-4 3 GHz CPU with 1 G memory. The performances of different methods are briefly summarized as follows:

- Methods IX and V achieve faster ART than other methods since the latter suffers from the problem of high dimensional feature space. One shot in Methods IX has only 8 dimension of motion vector, while one shot in method V has 162 dimension of HSV vector.
- Besides methods IX and V, our three methods using OM (methods I, II and III) are faster than other methods. In the three methods, the number of vertex in method III,

Table 6 The average retrieval time (ART) of one query for nine methods

One query	OM			IV	V	Methods based on [4]			IX
	I	II	III			VI	VII	VIII	
Average	6.5	5.9	4.8	6.8	4.5	15.5	12.0	11.3	0.7

in general, is less than that in methods I and II, and method I includes motion feature based on method II. Therefore method III is the fastest, while method II is faster than method I.

- In methods I and II, OM only needs less than 0.1 second for the average retrieval time (ART) of one query, while OM in method III is faster than methods I and II. This is because the number of vertex in method III is less than that of methods I and II. In fact, the main retrieval time is spent on other operations such as reading the feature files of shots.
- The three methods in [4] are slower than other methods. The reason is due to their feature representation and similarity measure. The method in [4] splits the image into a 5-by-5 grid. Each grid presents its color histogram in 125 dimensions. For each shot, the dimension of the color histogram is 3125 ($5 \times 5 \times 125$). So the time for similarity measure is relatively expensive.

Based on the findings in the above Sections 5.1, 5.2 and 5.3, overall, our proposed methods generally achieves better performances in terms of AR, ANMRR, precision, recall and the average retrieval time.

6 Conclusions

In this paper, a novel approach has been proposed for shot-based similarity measure by integrating color and motion features. In color similarity measure, OM is employed to compute the maximum weight of a bipartite graph as the similarity value between two shots. To improve the speed efficiency of OM, two improved algorithms are also proposed. In motion similarity measure, a motion histogram is constructed to represent every shot. The motion similarity is measured by the intersection between two motion histograms. The final similarity is based on the linear combination of color and motion similarity. Experimental results have indicated that the proposed approach achieves better performance than some existing methods.

Currently, the implementation of OM is based on Kuhn–Munkres algorithm which requires $O(n^4)$, where n is the total number of vertex in a weighted bipartite graph. Faster versions of OM algorithms exist in [14], for instance, OM can run in $O(n(m + n \log n))$, where m is the number of matching edges. In future, faster algorithm will be incorporated in the proposed approach for more efficient retrieval.

Acknowledgement The work described in this paper was fully supported by the National Natural Science Foundation of China under Grant No. 60503062, the Program for New Century Excellent Talents in University (NCET), and a grant from City University of Hong Kong (Project No. 7001804).

Appendix A

Average Normalized Modified Retrieval Rank (ANMRR)

Let Q as the number of queries and N as the number of items in a database. For a query q, $R(q)$ is defined as the set of relevant items in a database for q, and $NR(q)$ as the number of items in $R(q)$. Then, ANMRR is computed as

$$\mathrm{ANMRR} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\mathrm{MRR}(q)}{C(q)+0.5-0.5\times NR(q)} \tag{11}$$

where

$$C(q) = \min\left\{4\times NR(q), 2\times \max_{k=1}^{Q} NR(k)\right\}$$

$$NRR(q) = \frac{1}{NR(q)}\left\{\sum_{k=1}^{NR(q)} Rank(k,q)\right\} - 0.5 - \frac{NR(q)}{2}$$

The function $Rank(k,q)$ computes a value for an item which is retrieved as the kth most similar item to query q as

$$\mathrm{Rank}(k,q) = \begin{cases} k & if\, k \le C(q)\ and\ k\text{th item} \in R(q) \\ c(q)+1 & if\, k > C(q)\ and\ k\text{th item} \in R(q) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

The value of ANMRR will be in the range of [0.0, 1.0]. A lower value of ANMRR indicates a higher retrieval rate. Ideally, ANMRR=0 if the relevant items of all queries are appeared at the top rank lists.

Appendix B

Average Recall (AR)

Based on ANMRR, the AR is defined as

$$\mathrm{AR} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{RR(q)}{NR(q)} \tag{13}$$

where $RR(q)$ denotes the number of relevant items found in top $C(q)$ retrieved items.

References

1. Chen L, Chua TS (2001) A match and tiling approach to content-based video retrieval. International Conference on Multimedia and Expo
2. Deng Y, Manjunath BS (1997) Content-based search of video using color, texture and motion. International Conference on Image Processing 534–537

3. Fan J, Elmagarmid AK, Zhu X, Aref WG, Wu L (2004) Classview: hierarchical video shot classification, indexing, and accessing. IEEE Trans Multimedia 6(1):70–86
4. Hauptmann A, Chen M-Y, Christel M et al Confounded expectations: Informedia at TRECVID 2004. http://www-nlpir.nist.gov/projects/tvpubs/ tvpapers04/
5. Jain AK, Vailaya A, Wei X (1999) Query by video clip. Multimedia Syst 7:369–384
6. Lienhart R, Effelsberg W, Jain R (1998) VisualGREP: a systematic method to compare and retrieve video sequences. In: SPIE Conference on Storage and Retrieval for Image and Video Databases. pp 271–282
7. Lin T, Ngo CW, Zhang HJ et al (2001) Integrating color and spatial features for content-based video retrieval. In: IEEE International Conference on Image Processing (ICIP 2001). pp 592–595
8. Liu X, Zhuang Y, Pan Y (1999) A new approach to retrieve video by example video clip. ACM Multimedia Conference
9. MPEG video group (1999) Description of Core Experiments for MPEG-7 Color/Texture Descriptions. ISO/MPEGJTC1/SC29/WG11 MPEG98/M2819
10. Ngo CW, Pong TC, Chin RT (2001) Video partitioning by temporal slice coherency. IEEE Trans Circuits Syst Video Technol 11(8):941–953
11. Ngo CW, Pong TC, Zhang HJ (2002) Motion-based video representation for scene change detection. Int J Comput Vis 50(2):127–143 (Nov)
12. Over P, Kraaij W, Laneva T, Smeaton A, Buckland L TREC 2005 video retrieval evaluation introductions. http://www-nlpir.nist.gov/projects/tvpubs/ tv.pubs.org.html
13. Peng Y, Ngo CW (2006) Clip-based similarity measure for query-dependent clip retrieval and video summarization. IEEE Trans Circuits Syst Video Technol 16(5):612–627 (May)
14. Schrijver A (2003) Combinatorial optimization: Polyhedra and efficiency, vol A. Springer Heidelberg New York
15. Smeaton A, Laneva T TRECVID 2005: Search task. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html
16. Smeaton A, Over P, Arlandis J TRECVID-2004: Search task overview. http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html
17. Souvannavong F, Merialdo B, Huet B (2004) Latent semantic analysis for an effective region-based video shot retrieval system. In: The 6th ACM international workshop on multimedia information retrieval. New York, pp 243–250 (October)
18. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vis 7(1):11–32
19. Taskiran C, Chen J-Y, Albiol A, Torres L, Bouman CA, Delp EJ (2004) ViBE: a compressed video database structured for active browsing and search. IEEE Trans Multimedia 6(1):103–118
20. Wu Y, Zhuang Y, Pan Y (2000) Content-based video similarity model. In: ACM Multimedia Conference.
21. Xiao WS (1993) Graph theory and its algorithms. Aviation Industrial Press, Beijing
22. Yuan J, Duan L-Y, Tian Q, Wu C (2004) Fast and robust short video clip search using an index structure. In: The 6th ACM international workshop on multimedia information retrieval. New York, pp 61–68 (October)
23. Zhao L, Qi W, Li SZ et al (2000) "Key-frame extraction and shot retrieval using nearest feature line (NFL). In: ACM SIGMM international workshop on multimedia information retrieval.

Yuxin Peng received the Ph.D. degree in computer science and technology from Peking University, Beijing, China, in 2003.

He joined the Institute of Computer Science and Technology, Peking University, as an assistant professor in 2003 and was promoted to associate professor in 2005. From 2003 to 2004, he was a visiting scholar with the Department of Computer Science, City University of Hong Kong. His current research interests include content-based video retrieval, image processing and pattern recognition.

Chong-Wah Ngo (M'02) received his Ph.D. in Computer Science from the Hong Kong University of Science & Technology (HKUST) in 2000. He received his M.S. and B.S., both in Computer Engineering, from Nanyang Technological University of Singapore in 1996 and 1994 respectively.

Before joining City University of Hong Kong as assistant professor in Computer Science department in 2002, he was a postdoctoral scholar in Beckman Institute of University of Illinois in Urbana-Champaign (UIUC). He was also a visiting researcher of Microsoft Research Asia in 2002. CW Ngo's research interests include video computing, multimedia information retrieval, data mining and pattern recognition.

Jianguo Xiao is the head and professor in the Institute of Computer Science and Technology (ICST), Peking University, Beijing, China. He received the M.S. degree in computer science and technology from Peking University in 1988.

His research interests include image processing and information security. For the work and contributions in industry, he was the recipient of some awards in China, including national awards for science and technology achievement in 1995 (the first class) and 2006 (the second class), and Beijing's awards for science and technology achievement in 1993 (the special class) and 2005 (the first class).