On Clustering and Retrieval of Video Shots Through Temporal Slices Analysis

Chong-Wah Ngo, *Member, IEEE*, Ting-Chuen Pong, and Hong-Jiang Zhang, *Senior Member, IEEE*

***Abstract*—Based on the analysis of temporal slices, we propose novel approaches for clustering and retrieval of video shots. Temporal slices are a set of two–dimensional (2-D) images extracted along the time dimension of an image volume. They encode rich set of visual patterns for similarity measure. In this paper, we first demonstrate that tensor histogram features extracted from temporal slices are suitable for motion retrieval. Subsequently, we integrate both tensor and color histograms for constructing a two-level hierarchical clustering structure. Each cluster in the top level contains shots with similar color while each cluster in bottom level consists of shots with similar motion. The constructed structure is then used for the cluster-based retrieval. The proposed approaches are found to be useful particularly for sports games, where motion and color are important visual cues when searching and browsing the desired video shots.**

***Index Terms*—Hierarchical clustering, motion retrieval, temporal slices, tensor histogram.**

I. Introduction

CLUSTERING is a natural solution to abbreviate and organize the content of a video. A preview of the video content can simply be generated by showing a subset of clusters or the representative frames of each cluster. Similarly, retrieval can be performed in an efficient way since similar shots are indexed under the same cluster. Regardless of these advantages, general solutions for clustering video data is a hard problem. For certain applications, motion features dominate the clustering results; for others, visual cues such as color and texture are more important. Moreover, for certain types of applications, decoupling of camera and object motions ought to be done prior to clustering.

Video retrieval techniques, to date, are mostly extended directly or indirectly from image retrieval techniques. Examples include first selecting keyframes from shots and then extracting image features such as color and texture features from those keyframes for indexing and retrieval. The success from such extension, however, is doubtful since the spatio–temporal relationship among video frames is not fully exploited. Due to this consideration, recently more works have been dedicated to address this problem [3], [5], [6], [22], more specifically, to exploit and utilize the motion information along the temporal dimension for retrieval.

In this paper, we focus issues on clustering and retrieving the content of video shots. The major contributions are as follows.

- *Motion Retrieval*. We utilize various texture features such as tensor histogram [19], Gabor features [15] and the statistical features of co-occurrence matrix [9] extracted directly from temporal slices for motion retrieval. We compare the performance of these features with the histogram of MPEG motion vectors. Experimental results show that tensor histogram offers good compromise in term of retrieval accuracy and indexing speed.
- *Hierarchical Clustering*. By incorporating motion and color features, we propose a two-level hierarchical clustering structure to organize and index the content of video shots.
- *Cluster-Based vs. Cluster-Free Retrieval*. We investigate the effectiveness and efficiency of retrieving with (cluster-based) and without (cluster-free) clustering structure.

Our approach is based on the analysis and processing of patterns in temporal slice images. Temporal slices is a set of two–dimensional (2-D) images extracted along the time dimension of an image volume. They encode rich set of motion cues as oriented textures for shot similarity measure. Previous works on the analysis of temporal slices include visual motion model [1], [10], [23], [26], epipolar plane image analysis [2], surveillance monitoring [13], periodicity analysis [14], video partitioning [18], motion characterization and segmentation [19]. We contribute to this area of studies the utilization of texture features extracted directly from temporal slices for video clustering and motion retrieval.

We focus our attention for sport video domain, however, no specific domain knowledge is being utilized. We demonstrate that these videos are well represented as a two-level hierarchy. The top level is clustered by color features while the bottom level is clustered by motion features. The top level contains various clusters including wide-angle, medium-angle and close-up shots of players from different teams.[1] Each cluster can refer to a sport event. For instance, the wide-angle shots of a basketball video usually correspond to full court advances, while the wide-angle shots of a soccer video normally correspond to bird view scenes. The shots inside each cluster can be further partitioned according to their motion intensity. In this way, for example, the subcluster of a close-up shot can correspond either to "players running across the soccer field," or "players standing on the

Manuscript received March 29, 2001; revised March 7, 2002. This work was supported in part by RGC Grants HKUST661/95E, HKUST6072/97E, CityU1072/02E, SSRI99/00.EG11 and DAG01/02.EG16. The associate editor coordinating the review of this paper and approving it for publication was Prof. Alberto Del Bimbo.

C. W. Ngo is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: cwngo@cs.cityu.edu.hk).

T. C. Pong is with the Department of Computer Science, The Hong Kong University of Science & Technology, Kowloon, Hong Kong (e-mail: tcpong@cs.ust.hk).

H. J. Zhang is with Microsoft Research Asia, Beijing 100 080, China (e-mail: hjzhang@microsoft.com).

Digital Object Identifier 10.1109/TMM.2002.802022

[1]The classification of wide-angle, medium-angle and close-up shots are roughly based on the distance between the camera lens and the targeted scene.

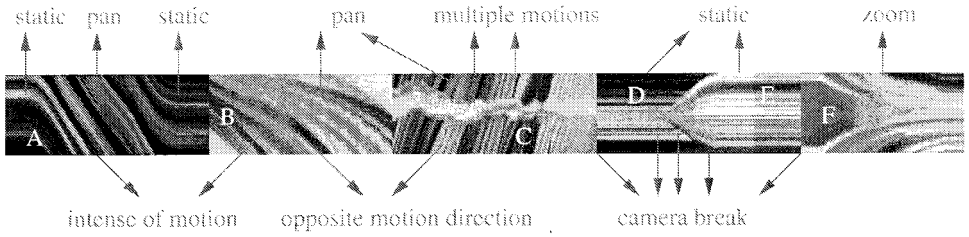


Fig. 1. Patterns in a spatio–temporal slice.

field." Such organization facilitates not only video browsing and retrieval, but also some high-level video processing tasks. For instance, to perform player recognition, only those shots in the cluster that correspond to close-up shots of players are picked up for processing. To perform motion-based background reconstruction, the subcluster corresponds to "players running across the soccer field" is further selected for processing.

For motion retrieval, unlike [5], [6], neither object tracking nor motion decomposition is performed for explicitly isolating dominant camera motion from object motion prior to feature extraction. This is because camera motion in part plays an important role in conveying the content of sport videos. In contrast to other video sources where the motion of camera is unrestricted, sport videos are usually captured by several fixed cameras that are mounted in the stand. These camera motion are mostly regular and driven by the pace of sport games or the events that are taken place on spot. Most importantly, when coupling with the object motion of a particular event in sport videos, a unique texture pattern can always be observed in temporal slices.

The remaining of paper is organized as follows. Section II describes various texture patterns formed in temporal slices due to different activities in sports. Section III proposes various methods to extract texture features from temporal slices for motion retrieval. Section IV starts by proposing a two-level hierarchical clustering structure. Issues on cluster-based vs. cluster-free retrieval are then investigated. Section V concludes our proposed works.

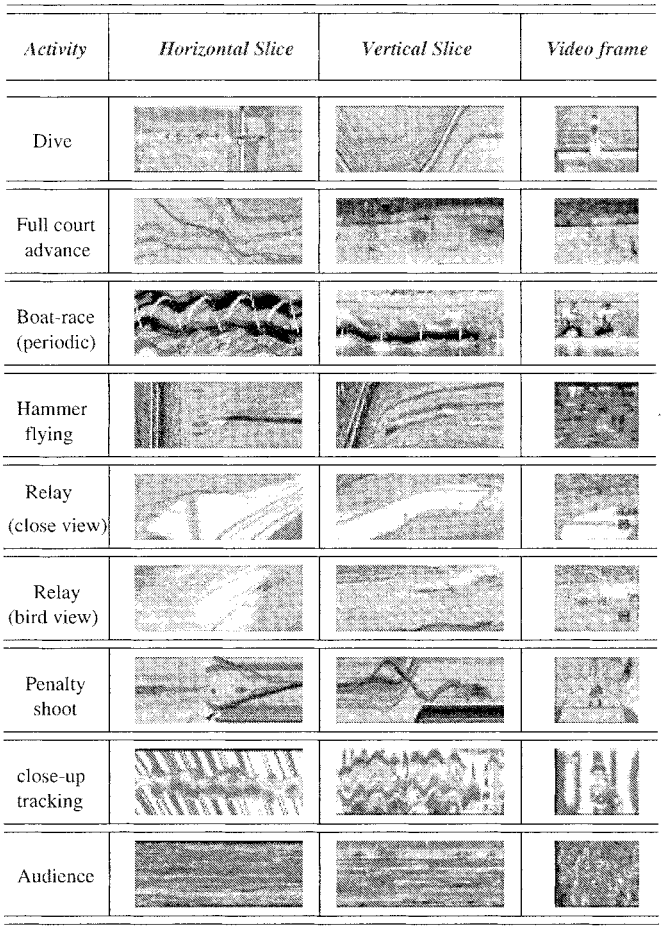


Fig. 2. Patterns in both horizontal and vertical slices.

II. Spatio–Temporal Pattern Analysis

Temporal slices are a set of 2-D images in an image volume with one dimension in t, and the other in x or y, for instance. Fig. 1 shows a temporal slice extracted from a video composed of six shots; the horizontal axis is t, while the vertical axis is x. A temporal slice, by first impression, is composed of color and texture components. On one hand, the discontinuity of color and texture infers the occurrence of a new event; on the other hand, the orientation of texture depicts camera and object motions. The patterns in temporal slices, perceptibly, can be exploited for video partitioning [18], motion characterization and segmentation [19]. As seen in Fig. 1, motion can be inferred directly from the texture pattern. For instance, the horizontal lines denote static motion; the slanted lines depict panning; the lines expanded in V-shape pattern depict camera zooming; two dissimilar texture patterns in a shot indicate multiple motions.

A. Patterns in Temporal Slices

Fig. 2 further shows the patterns of various activities in the horizontal ($x-t$ dimensions) and vertical ($y-t$ dimensions) slices. It is worthwhile to observe the following details.

- For diving, since the motion proceeds in vertical direction, the vertical slices depict camera tilting while the horizontal slices explore panoramic information. A full court advance in basket videos, on the other hand, has the horizontal slices depict camera panning while the vertical slices explore panoramic information.
- The periodic motion in the boat-race shot is indicated in the horizontal slices.
- The camera motion which tracks a flying hammer in a parabolic-like direction is depicted in the slanted lines of vertical slices.

- The close-view and bird-view of a scene will also give two different perception patterns on temporal slices. The delineation of scene is vividly observed in the former case, however, is smeared in the later case.
- The incoherent motion of camera and objects is also depicted in temporal slices. For the penalty shot in basket videos, the trajectory of ball is weakly seen in the vertical slices when the camera tilts. For the close-up tracking shot, the nonrigid motion of a player is seen in both horizontal and vertical slices when the camera pans.
- For the audience scene, the temporal slices show stationary motion pattern with random noise due to audiences' movement.

As seen in the examples, the diversities of texture patterns are encoded in both horizontal and vertical slices by different sport activities. Intuitively, motion retrieval can be done by extracting texture patterns directly from slices. In principle, all temporal slices, both horizontal and vertical, should be processed for motion pattern analysis. Nevertheless, computational time can be saved (while better analytical results can be acquired) if a subset of slices are randomly (or intelligently) selected for processing. In this paper, unless being stated, all slices are used for feature extraction.

III. Motion Retrieval

Motion features that have been used for retrieval include the motion trajectories of objects [5], principle components of MPEG motion vectors [22], and temporal texture [6]. In this section, we propose new ways of computing and extracting temporal texture. Temporal texture was primarily proposed by Polana and Nelson to describe the dynamic of temporal events [17]. As image texture, temporal texture can be modeled as co-occurrence matrix [3], [17], autoregressive model [24], wold decomposition [14] and Gibbs random field [6]. Except Fablet and Bouthemy, who described the use of temporal texture for video retrieval [6], this feature is mostly utilized for recognizing complex dynamic motion such as rivers and crowds [3], [17], [24], and detecting periodic motion such as walking and swimming [14].

For most approaches [6], [17], the input to temporal texture is optical flow or normal flow field. In other words, motion information need to be explicitly computed before the generation of temporal texture. Consequently, the effectiveness of the computed temporal feature is dependent on the reliability of input motion information. Unfortunately, motion information such as optical flow is not only computationally expensive but also noise sensitive. Our proposed methods, with contrary to these approaches, computes temporal texture by taking the gray-level information of temporal slices as input. In this section, we present our proposed methods to extract and represent the texture patterns in slices as tensor histogram [19], Gabor feature [15], and co-occurrence matrix [9] for motion retrieval.

A. Feature Extraction

For computational and storage efficiency, all the features are extracted from the temporal slices that are obtained directly from the compressed video domain. Slices can be obtained from the DC image volume which is easily constructed by extracting the DC components[2] of MPEG video.

1) Tensor Histogram: Tensor histogram encodes the distribution of local orientation in temporal slices. It is computed based on the structure tensor introduced in [7], [13] to estimate the orientations of slices. This feature has been used by Ngo *et al.* for motion characterization and segmentation [19].

The structural tensor Γ of slice $\mathbf{H}$ can be expressed as

$$\Gamma = \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix} = \begin{bmatrix} \sum_w \mathbf{H}_x^2 & \sum_w \mathbf{H}_x\mathbf{H}_t \\ \sum_w \mathbf{H}_x\mathbf{H}_t & \sum_w \mathbf{H}_t^2 \end{bmatrix} \tag{1}$$

where $\mathbf{H}_x$ and $\mathbf{H}_t$ are partial derivatives along the spatial and temporal dimensions respectively. The window of support w is set to 3×3 throughout the experiments. The rotation angle θ of Γ indicates the direction of a gray level change in w. Rotating the principle axes of Γ by θ, we have

$$\mathbf{R}\begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix}\mathbf{R}^T = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_t \end{bmatrix} \tag{2}$$

where

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

From (2), since we have three equations with three unknowns, θ can be solved and expressed as

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mathbf{J}_{xt}}{\mathbf{J}_{xx} - \mathbf{J}_{tt}}. \tag{3}$$

The local orientation ϕ of a w in slices is computed as

$$\phi = \begin{cases} \theta - \dfrac{\pi}{2} & \theta > 0, \\ \theta + \dfrac{\pi}{2}, & \text{otherwise} \end{cases} \qquad \phi = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]. \tag{4}$$

It is useful to add in a certainty measure to describe how well ϕ approximates the local orientation of w. The certainty c is estimated as

$$c = \frac{(\mathbf{J}_{xx} - \mathbf{J}_{tt})^2 + 4\mathbf{J}_{xt}^2}{(\mathbf{J}_{xx} + \mathbf{J}_{tt})^2} = \left(\frac{\lambda_x - \lambda_t}{\lambda_x + \lambda_t}\right)^2 \tag{5}$$

and $c = [0, 1]$. For an ideal local orientation, $c = 1$ when either $\lambda_x = 0$ or $\lambda_t = 0$. For an isotropic structure, i.e., $\lambda_x = \lambda_t$, $c = 0$.

The distribution of local orientations across time inherently reflects the motion trajectories in an image volume. A 2-D tensor histogram $\mathbf{M}(\phi, t)$ with the dimensions as a one–dimensional (1-D) orientation histogram and time respectively, can be constructed to model the distribution. Mathematically, the histogram can be expressed as

$$\mathbf{M}(\phi, t) = \sum_{\Omega(\phi, t)} c(\Omega) \tag{6}$$

[2]The algorithm introduced by Yeo and Liu [27] is applied to estimate DC components from P-frames and B-frames.

where $\Omega(\phi, t) = \{\mathbf{H}(x, t) | \Gamma(x, t) = \phi\}$ which means that each pixel in slices votes for the bin (ϕ, t) with the certainty value c.

For motion retrieval, a 1-D tensor histogram $\mathcal{M}(k)$ is computed directly by

$$\mathcal{M}(k) = \frac{1}{n}\left\{\sum_{\phi'}\sum_{t} M(\phi', t)\right\} \qquad \forall_{\phi'}\{\mathcal{Q}(\phi') = k\} \quad (7)$$

where $\mathcal{Q}(\phi')$ is a quantization function, and $k = \{1, 2, \ldots, 8\}$ represents a quantized level. The histogram is uniformly quantized into 8 bins with each bin has a range $\pi/8$. The computed motion features describe the motion intensity and direction of shots. In our experiment, the tensor histograms of both horizontal and vertical slices are used for feature computation. As a result, the total feature vector length is 16.

2) Gabor Feature: Gabor feature is frequently used for browsing and retrieval of texture images, and have been shown to give good results [15]. A Gabor filter $g(x, t)$ can be written as

$$G(x, t) = \left(\frac{1}{2\pi\sigma_x\sigma_t}\right)\exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{t^2}{\sigma_t^2}\right)\right\}\exp\{2\pi jWx\} \quad (8)$$

where σ_x and σ_t are smoothing parameters, $j = \sqrt{-1}$, $W = \sqrt{u^2 + v^2}$ and (u, v) is the center of the desired frequency. A self-similar filter $G_{\theta\mathcal{S}}(x, t)$ can be obtained by the appropriate rotation θ and scaling $\mathcal{S}$ of $G(x, t)$ [4], [15].

The Gabor filtered image of a slice $\mathbf{H}$ is

$$\hat{\mathbf{H}}_{\theta\mathcal{S}} = \mathbf{H} * G_{\theta\mathcal{S}} \quad (9)$$

where $*$ is a convolution operator. A feature vector is constructed by using the mean $\mu_{\theta\mathcal{S}}$ and the standard deviation $\sigma_{\theta\mathcal{S}}$ of all $\hat{\mathbf{H}}_{\theta\mathcal{S}}$ as components. In the experiment, $\theta = 6$ and $\mathcal{S} = 2$. The resulting feature vector has length $6 \times 2 \times 2 \times 2 = 48$ in the following form

$$\underbrace{[\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots \mu_{51}, \sigma_{51}}_{\text{for horizontal slices}},\ \underbrace{\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \ldots \mu_{51}, \sigma_{51}]}_{\text{for vertical slices}}.$$

3) Co-Occurrence Matrix: Gray level co-occurrence matrix is frequently utilized to describe image texture [9]. The second order statistical features can be computed directly from a matrix to characterize the spatial relationships of gray level properties. The co-occurrence matrix of a slice can be represented by $P(i, j; d, \theta)$. It specifies the frequencies of two neighboring pixels separated by distance d at orientation θ in the temporal slices, one with gray level i and the other with gray level j.

In our experiment, $d = \{1, 2, 3, 4, 5\}$ and $\theta = \{-45°, 0°, 45°\}$. The co-occurrence matrices of horizontal and vertical slices are computed, summed and normalized separately, hence, there are thirty matrices used to model the spatial relationships of slices. The smoothness features $Sm(d, \theta)$ and contrast features $Con(d, \theta)$ are then computed directly from these matrices by

$$Sm(d, \theta) = \sum_i \sum_j P^2(i, j; d, \theta) \quad (10)$$

$$Con(d, \theta) = \sum_i \sum_j (i - j)^2 P(i, j; d, \theta). \quad (11)$$

The resulting feature vector has length $15 \times 2 \times 2 = 60$.

B. Distance Measure

The L_1 and L_2 norms are two of the most frequently used distance metrics for comparing two feature vectors. In practice, however, L_1 norm performs better than L_2 norm since it is more robust to outliers [20]. Furthermore, L_1 norm is more computationally efficient and robust. The distance between two nonzero length feature vectors $\mathcal{F}$ and $\mathcal{F}'$ is computed as

$$D(\mathcal{F}, \mathcal{F}') = \frac{1}{Z(\mathcal{F}, \mathcal{F}')}\left\{\sum_{i=1}^{n} |\mathcal{F}(i) - \mathcal{F}'(i)|^k\right\}^{1/k} \quad (12)$$

where

$$Z(\mathcal{F}, \mathcal{F}') = \sum_{i=1}^{n} \mathcal{F}(i) + \sum_{i=1}^{n} \mathcal{F}'(i) \quad (13)$$

is a normalizing function. In (12), $k = 1$ for L_1 norm and $k = 2$ for L_2 norm. For tensor histogram, we use L_1 norm as distance measure. For Gabor feature and co-occurrence matrix, because the range of different feature components can significantly vary, we use the following distance measure

$$D(\mathcal{F}, \mathcal{F}') = \sum_{i=1}^{n} \left|\frac{\mathcal{F}(i) - \mathcal{F}'(i)}{\alpha(i)}\right| \quad (14)$$

where $\alpha(i)$ is the standard deviation of the ith feature component over the entire database.

C. Experiments

We conduct experiments on basketball, soccer, and TV sport video databases. The basketball video consists of 76 shots (approximately 18 000 frames); the soccer video consists of 404 shots (approximately 100 000 frames); and the TV sport video consists of 180 shots (about 37 000 frames). We adopt RP (recall-precision) and ANMRR (average normalized modified retrieval rank) to evaluate the retrieval accuracy. Both performance measures are popularly used in the current literature [16]. The values of RP and ANMRR range between [0, 1]. A high value of RP denotes the superior ability in presenting relevant shots in the top retrievals, while a low value of ANMRR indicates the high retrieval rate with relevant shots ranked at the top (see Appendixes A and B for details).

Besides comparing the performance among the features extracted from temporal slices, we also contrast their retrieval accuracy with MPEG motion vectors which are computed through the block-based motion estimation algorithm. Only motion vectors from P-frames are used and they are represented by a histogram that is composed of eight bins. Each bin corresponds to

TABLE I
RETRIEVAL ON BASKETBALL DATABASE

Approach	Mean Precision	ANMRR
Tensor histogram	0.553*	0.399*
Gabor	0.502	0.431
Co-occurrence matrix	0.426	0.543
MPEG motion vector	0.470	0.498

*The mark * indicates the best performance.*

TABLE II
RETRIEVAL ON SOCCER DATABASE

Approach	Mean Precision	ANMRR
Tensor histogram	0.457*	0.377*
Gabor	0.435	0.430
Co-occurrence matrix	0.400	0.492
MPEG motion vector	0.321	0.590

*The mark * indicates the best performance.*

TABLE III
RETRIEVAL ON SPORT DATABASE

Approach	Mean Precision	ANMRR
Tensor histogram	0.511*	0.456*
Gabor	0.482	0.481
Co-occurrence matrix	0.393	0.577
MPEG motion vector	0.413	0.557

*The mark * indicates the best performance.*

one of the eight neighborhood directions in the discrete space. To take motion intensity into account, each bin contains the total length, instead of frequency, of the motion vectors having same direction. Similar to tensor histogram, L_1 norm is employed for distance measure.

1) Retrieval Accuracy: For each database, we manually annotate and categorize the video content for performance evaluation. A number of queries which are checked to have unique answers are selected from these categories for testing. These queries are executed on the database and not on the categorized data. The retrieval results will be compared against the manually categorized data and evaluated by RP and ANMRR.

In the basketball database, the shots are categorized into full-court-advance (FCA), close-up shots of player, penalty shots, shooting shots, and audience scene. Such categorization is based not only on the semantic events of basketball videos, but also mainly based on the motion content of shots. For instance, a FCA shot is usually associated with a camera that is panned toward the direction when the ball is being advanced from one end of the court to the other; a penalty shot is accompanied with a camera that tilt up when the ball is shot; a shooting shot is normally associated with a zoom to emphasize the moment of shooting; while a audience scene which is captured by a static camera exhibits random motion patterns due to audiences' cheering. In addition, the close-up of players are further classified into players moving to the left, players moving to the right, and players with no motion.

In this database, twenty queries that are manually checked to have good answers are picked for testing. The retrieval performance is given in Table I. The stars marked in the table entries indicate the best experimental results among all the tested approaches. In this experiment, tensor histogram outperforms other approaches in term of mean precision and ANMRR. The mean precision is the average of precision values at different recall levels.

The categorization of the soccer database is similar to the basketball database. The shots in the database are classified into bird views, medium shots, close-up shots of players, shooting scenes and audience scenes. A total of fifty three queries from these categories are selected for testing. The experimental results are shown in Table II. Similarly, tensor histogram outperforms other approaches in term of mean precision and ANMRR.

The sport database contains a diversity of sport games including diving, golf and race. We categorize the shots according to the type of sport games. Some games are further categorized into bird view or close-up shots. The close-up shots are also categorized into tracking or stationary shots. A total of 124 shots from these categories are selected for testing. The experimental results, as shown in Table III, indicate that tensor histogram on average gives the best performance. To better illustrate the superiority of features extracted from temporal slices, Table IV further compares its performance with MPEG motion vector on various sport events. Among the twelve sport activities, temporal texture feature gives the best performance for nine events. In Fig. 3(a) and (b), two examples of motion retrieval are shown together with the results given by tensor histogram and MPEG motion vectors. These examples demonstrate the superior capability of tensor histogram in ranking and retrieving the shots that are similar to the queries. Histogram of MPEG motion vectors, in contrast, retrieves shots that are perceptually very different from the queries.

2) Speed Efficiency: Table V compares the performance efficiency in term of the feature vector length and the feature extraction time (second per image frame). Because all horizontal and vertical slices are used for feature extraction, the features extracted from temporal slices are not as efficient as motion vector histogram. Among these features, tensor histogram is computationally superior to other two approaches. It is about eleven times faster than Gabor feature and about two times faster than co-occurrence matrix.

3) Number of Slices Used: In most cases, temporal slices in an image volume are highly correlated. For computational efficiency, probably only a few selected slices instead of a whole image volume are sufficient for motion feature extraction. We conduct an experiment by uniformly sampling the temporal slices and picking only a subset of slices for processing. At the extreme level, only two slices (i.e., a vertical slice and a horizontal slice located at the center of an image volume) are selected.

Table VI shows the experimental results of the tensor histogram approach. Each time when the number of slices are reduced by half, the speed of feature extraction is improved by approximately 1.2 to 1.7 times without significantly degrading

TABLE IV
EVENT RETRIEVAL ON SPORT DATABASE

Event	ANMRR		Number of queries
	Temporal texture feature	MPEG motion vector	
Diving	0.241*	0.410	31
Golf-shooting	0.094*	0.468	6
Golf-flying	0.108*	0.530	8
Hammer-throwing	0.664	0.627*	5
Hammer-flying	0.415*	0.672	5
Relay (close-up view)	0.531	0.467*	18
Relay (bird-view)	0.491*	0.636	8
Boat-race (close-up view)	0.475	0.434*	3
Boat-race (bird-view)	0.052*	0.453	4
Static motion	0.534*	0.553	12
People tracking	0.542*	0.648	18
Audience	0.515*	0.780	6

*The ANMRR values of temporal texture feature are represented by the ANMRR values of tensor histogram, Gabor or co-occurrence matrix with best performance. The mark * indicates the better performance.*

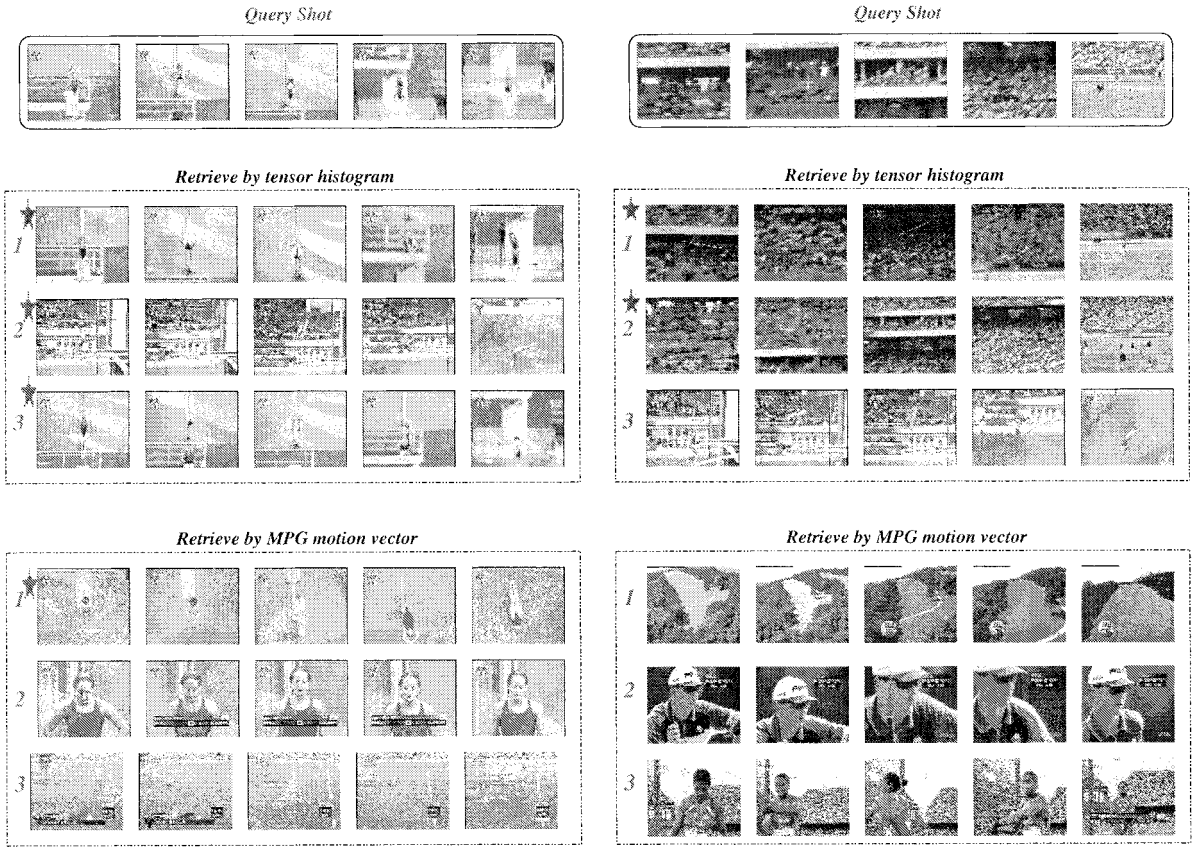


Fig. 3. Motion retrieval examples (show the three most similar shots, stars mark the correct answers).

the retrieval performance. Appropriate reduction of slices will generally improve the retrieval accuracy. This may due to the elimination of some slices that contain homogeneous regions or image noise during feature extraction. It should be noted that, for all the tested database, when only two slices are used, the processing speed of the tensor histogram is comparable to MPEG motion vector histogram while the retrieval accuracy is still superior to all other approaches.

TABLE V
SPEED EFFICIENCY OF VARIOUS APPROACHES

Approach	Feature Vector Length	Feature Extraction (sec)
Tensor histogram	16	0.072
Gabor	48	0.791
Co-occurrence matrix	60	0.130
MPEG motion vector	8	0.017*

*The mark * indicates the best performance.*

TABLE VI
PERFORMANCE OF TENSOR HISTOGRAM

Number of slices	Feature Extraction time (sec)	ANMRR		
		Basketball	Soccer	Sport
All	0.072	0.399	0.393	0.456
Half	0.042	0.399	0.392*	0.453
One-third	0.033	0.397	0.392*	0.450
One-quarter	0.028	0.392*	0.394	0.448*
Two	0.015*	0.416	0.416	0.471

*The mark * indicates the best performance.*

D. Discussion

Three new temporal texture features based on the analysis of temporal slices have been presented and applied to motion retrieval in sport video databases. Among the proposed features, tensor histogram is empirically found to be superior to other features in term of retrieval accuracy and speed efficiency. Furthermore, the computational time of tensor histogram can be as fast as the histogram of MPEG motion vectors by reducing the number of slices being processed without significantly degrading the retrieval performance.

The main advantage of tensor histogram is that it can accurately encode the visual patterns inherent in temporal slices. In contrast to MPEG motion vector histogram which does not take into account the texture information of image region, tensor histogram takes the certainty measure [(5)] into consideration. As a result, less weights are contributed by nontexture regions (the weight of homogenous region is zero) compared to texture region, as indicated in (6). When parts of video shots are occupied by nontexture regions (this situation happens frequently in video clips with diving, golf-shooting and bird view scenes), tensor histogram performs better than MPEG motion vectors.

Nevertheless, currently only the visual patterns of 2-D horizontal and vertical slices are considered, tensor histogram suffers from the imprecise description of the rotational and diagonal motions. Typical examples include the close-up relay shots where racers approach from top left-hand screen to bottom right-hand screen, the hammer-throwing shots where the rotational motion crosses diagonally through most of the temporal slices. Such problems, however, can be handled by modeling the relationship among temporal slices, or simply by encoding the texture patterns of diagonal slices.

IV. CLUSTERING AND RETRIEVAL

In this section, we further utilize the results of motion retrieval shown in Section III for clustering and retrieval. A hierarchical structure which provides indexing scheme for retrieval is constructed by incorporating motion and color features. Since tensor histogram offers good compromise between retrieval effectiveness and speed efficiency, it is utilized as motion features. In addition, based on the experimental results indicated in Table VI, we use only one-quarter of horizontal and vertical slices, instead of the whole image volume, for motion feature computation. Besides clustering, the fundamental difference between cluster-based and cluster-free retrieval is studied. The performance of motion retrieval, color retrieval, and the integration of motion and color retrieval are compared and contrasted.

For clustering algorithms, basically they can be grouped into two categories: partitional and hierarchical [12]. Hanjalic and Zhang [8] have introduced a partitional clustering of video data by utilizing the color features of selected keyframes. Here, we introduce a two-level hierarchical clustering algorithm by integrating both color and motion features. The proposed clustering algorithm is unsupervised, the number of clusters is determined automatically by the cluster validity analysis [12]. Supervised version of clustering algorithms by Hidden Markov Models can be found in [11], [22], and by decision rules can be found in [25].

A. Feature Extraction

We present methods to extract motion and color features directly from shots. Both motion and color features are represented as histograms, since this representation is effective and inexpensive for clustering and retrieval.

1) Motion Feature: Unlike previous section, motion directional information computed by structure tensor is not encoded as we target our applications for sport activities that involve two teams such as basketball and soccer. Motion direction in these activities does not provide additional clue for clustering. For instance, it is not useful to put players moving in different directions into distinct clusters.

With reference to (7), the quantization function is modified to $\mathcal{Q}(\phi') = (8 \times |\phi'|)/\pi$ where $\phi' = [-(\pi/2), \pi/2]$. Since $|\phi'| \geq 0$, the motion feature is directionless and the quantized level $k = \{1, 2, 3, 4\}$. The resulting feature vector length is $4 \times 2 = 8$.

2) Color Feature: The color feature of a shot, represented as a three-dimensional color histogram, is computed directly in the YUV color space of its DC image sequence. A color histogram describes the global color distribution in a shot. It is easy to compute and is insensitive to small changes in viewing positions and partial occlusion. As a feature vector for clustering and retrieval, it is susceptible to false alarms. In our experiments, each color channel of YUV is uniformly quantized to four bins, results in a 64 dimensional color feature vector. Each color histogram will be further normalized by the number of frames in a shot.

B. Hierarchical Clustering

We employ a two-level hierarchical clustering approach to group shots with similar color and motion. The algorithm is implemented in a top-down fashion, where color features are utilized at the top level, while motion features are used at the bottom level. At the top level, the color feature space is partitioned to k_c clusters. At the bottom level, each k_c cluster is further partitioned into k_m clusters.

1) K-Mean Algorithm: The k-mean algorithm is the most frequently used clustering algorithm due to its simplicity and efficiency. The algorithm is employed to cluster shots at each level of hierarchy independently. The k-mean algorithm is implemented as

- *Step 1*: Choose $\mu_1, \mu_2, \ldots, \mu_k$ as initial cluster centroids.
- *Step 2*: Classify each feature $\mathcal{F}$ to the cluster $\hat{p}$ with the smallest distance

$$\hat{p} = \arg\min_{1 \le j \le k} D(\mathcal{F}, \mu_j). \tag{15}$$

- *Step 3*: Based on the classification, update cluster centroids as

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathcal{F}_i^{(j)} \tag{16}$$

where n_j is the number of shots in cluster j, and $\mathcal{F}_i^{(j)}$ is the ith feature vector in cluster j.

- *Step 4*: If any cluster centroid changes value, goto *Step 2*; otherwise stop.

2) Cluster Validity: The number of clusters k needs to be explicitly specified for the k-mean algorithm. However, in most cases, k is not exactly known in advance. In order to find an optimal number of clusters, we have employed the cluster validity analysis [12]. The intuition is to find clusters that minimize intra-cluster distance while maximize inter-cluster distance. The cluster separation measure $\rho(k)$ is defined as

$$\rho(k) = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \le n \le k} \left\{ \frac{\eta_i + \eta_j}{\xi_{ij}} \right\} \tag{17}$$

where

$$\eta_j = \frac{1}{n_j} \sum_{i=1}^{n_j} D\left(\mathcal{F}_i^{(j)}, \mu_j\right) \tag{18}$$

$$\xi_{ij} = D(\mu_i, \mu_j) \tag{19}$$

η_j is the intra-cluster distance of cluster j, while ξ_{ij} is the inter-cluster distance of clusters i and j. In the experiments, the optimal number of clusters $\hat{k}$ is selected as

$$\hat{k} = \min_{1 \le k \le 10} \rho(k). \tag{20}$$

In other words, the k-mean algorithm is tested for $k = \{1, 2, \ldots, 10\}$, and the one which gives the lowest value of $\rho(k)$ is chosen.

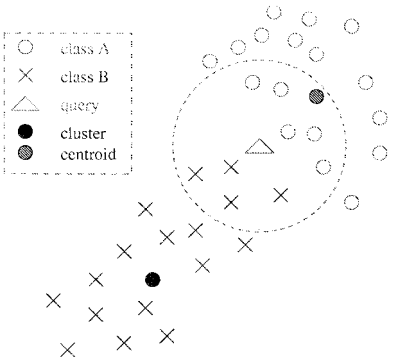


Fig. 4. When a query locates at the boundary of two classes, the retrieval results of the cluster-based and cluster-free approach will be quite different.

C. Retrieval

Given a query represented by low-level features, motion and color in our case, a retrieval system returns a set of items sorting in ascending order according to their respective distances to the query. This is normally referred to as a k nearest neighbor (KNN) search problem, which is actively studied by both computational geometry and multimedia retrieval communities. By coupling clustering issues with retrieval problems, the clustering structure, on one hand, inherently provides an indexing scheme for retrieval, while on the other hand, intuitively speed up the retrieval time. We refer to this issue as a cluster-based retrieval problem. The fundamental difference between cluster-based and cluster-free retrieval are illustrated in Fig. 4. Suppose a query is located at the boundary of two classes, cluster-free retrieval will include items from both classes, i.e., items inside the dotted circle in Fig. 4, at the top of a ranked list. In contrast, clustered-based retrieval compares the distance between the query and each cluster centroid, and ranks the items whose cluster centroid is nearer to the query at the top of a list.

1) Cluster-Based Retrieval: In a hierarchical clustering structure, a centroid at the top level represents the color characteristics of a cluster, while a centroid at the bottom level represents the motion characteristics of a cluster. During retrieval, cluster centroids at the top level of hierarchy are first compared with the color feature of a query. A cluster with the nearest centroid is first located. Then, its subclusters in the bottom level are further compared with the query. The items in one of these subclusters whose centroid is the nearest to the motion feature of the query, are sorted in ascending order of their distance to the query, and put accordingly at the top of a ranked list. The retrieval is processed in a depth-first-search-like manner, meaning that after all subclusters of the most similar cluster are sorted, the next similar cluster is handled in the same way. This process is repeated until the few most similar or all clusters are visited.

2) Cluster-Free Retrieval: If a clustering structure is not available, we can merge the ranked lists given by both motion and color features. A straightforward way is to linearly weight the distance measures given by both features. Denote $\langle \mathcal{M}, \mathcal{C} \rangle$

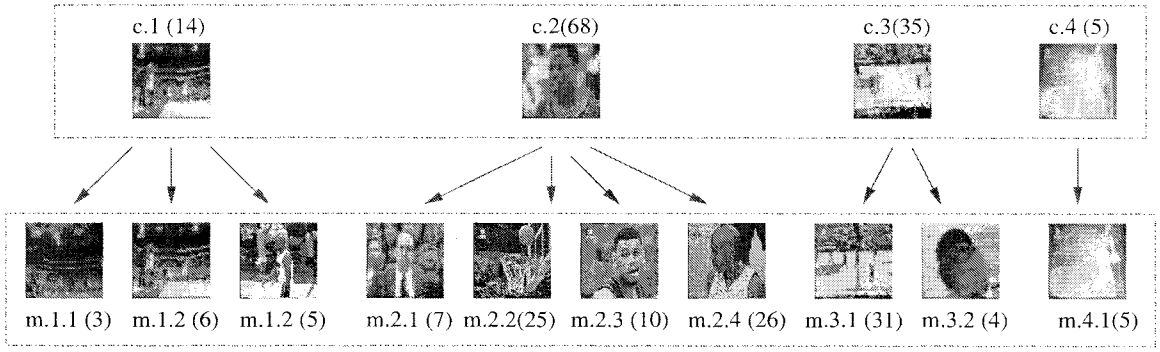


Fig. 5. Clustering result of basketball video by using L_1 norm as distance measure. $X(Y)$: X is cluster label and Y is the number of shots in a cluster.

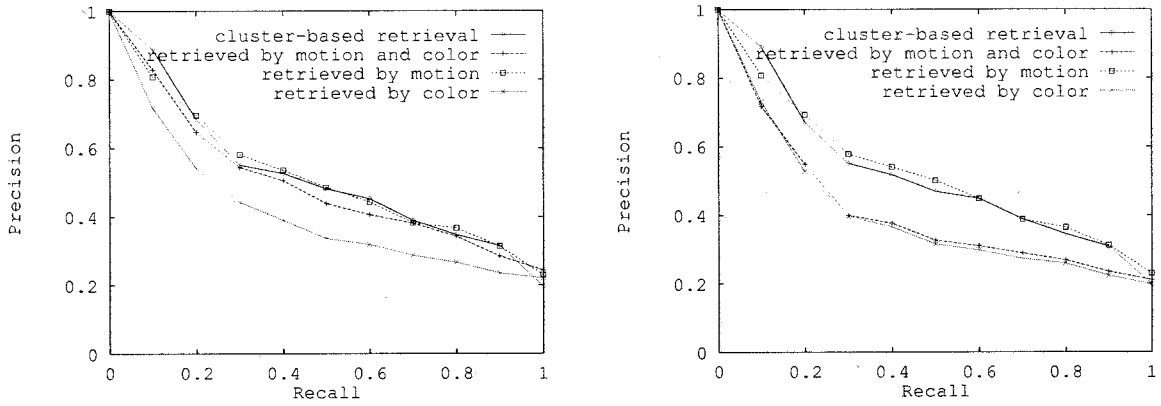


Fig. 6. Recall and precision curves for basketball video. (a) L_1 norm distance measure and (b) L_2 norm distance measure.

and $\langle \mathcal{M}', \mathcal{C}' \rangle$ as two pairs of motion and color feature vectors, we have

$$\mathbf{D}(\langle \mathcal{M}, \mathcal{C} \rangle, \langle \mathcal{M}', \mathcal{C}' \rangle) = \alpha_{\mathcal{M}} D(\mathcal{M}, \mathcal{M}') + \alpha_{\mathcal{C}} D(\mathcal{C}, \mathcal{C}') \tag{21}$$

where $\alpha_{\mathcal{M}}$ and $\alpha_{\mathcal{C}}$ are weights, and $\alpha_{\mathcal{M}} + \alpha_{\mathcal{C}} = 1.0$. To equally weight both features, we set $\alpha_{\mathcal{M}} = \alpha_{\mathcal{C}} = 0.5$.

D. Experiments

To test the effectiveness of the proposed clustering and retrieval approach, we conduct experiments on both the basketball and soccer videos. The basketball video composed of 122 shots (approximately 24 000 frames) while the soccer video is composed of 404 shots (approximately 100 000 frames). The tested videos are first partitioned into shots and then 2-D tensor histograms are computed for each shot. For clustering, the motion and color features are extracted, respectively, from the 2-D tensor histograms and image volumes of shots. In addition, the performance of both the cluster-based and the cluster-free retrieval approach is investigated. For the cluster-free retrieval, we examine also the retrieval by motion feature, retrieval by color feature, and retrieval by both color and motion features. For all the tested approaches, we investigate the retrieval performance of employing L_1 norm and L_2 norm as distance measure. For cluster-based retrieval, both clustering structures by L_1 norm and L_2 norm are constructed for retrieval. The retrieval performance is evaluated in terms of recall and precision.

1) Basketball Video: Fig. 5 depicts the clustering results by using L_1 norm as the distance measure. Shots that are nearest to cluster centroids are shown in the figure to represent clusters. The top level has four clusters, while the bottom level consists of ten subclusters. By manual investigation of the clustering results, we summarize the characteristics of each cluster as follows.

- Cluster *c.1* mostly consists of penalty shots, audience scene and few close up shots.
- Cluster *c.2* basically consists of players from both teams. The players can not be classified according to their teams due to the unsegmented cluttered background. The subclusters *m.2.1* and *m.2.2* contain shots with slight motion, while the subclusters *m.2.3* and *m.2.4* consist of shots that track players along the court. In additions, most shots in *m.2.2* and *m.2.4* are contaminated with camera flashing.
- Cluster *c.3* has mainly the FCA shots. These shots are grouped accordingly in the subcluster *m.3.1* by motion features. The subcluster *m.3.2* consists of four non-FCA shots which are classified incorrectly by color features.
- Cluster *c.4* consists of the logo sequence which is appeared prior to the replay of slow motion sequence.

To evaluate the retrieval performance, a query set which consists of 25 queries are manually picked and checked to have good answers. They consist of close up of players from different teams, FCA, penalty and shooting shots. Players from different teams are placed in different classes. Similarly, players captured

by different camera motions (e.g., track and zoom) are put in different classes.

Fig. 6(a) shows the recall-precision of the tested approaches by using L_1 norm as distance measure, while Fig. 6(b) shows the recall-precision by using L_2 norm as distance measure. Table VII summarizes the 11-point average precision values of various approaches. As indicated from the experimental results, the retrieval performance based on L_1 norm distance measure is robuster than L_2 norm. Throughout the experiments, motion features alone give better retrieval performance compared with color features. The retrieval performance of cluster-based and cluster-free retrieval by motion, in general, does not show significantly different results. Nevertheless, the merit of cluster-based retrieval is that it offers higher precision at recall level less than or equal to 0.1. This is generally true for almost all the queries that we have tested. We expect this desirable characteristic could provide a good start for relevancy feedback mechanism [21] since users can quickly identify relevant video clips by browsing a small set of retrieved items. At recall level equals to 1.0, however, the performance of *cluster-based approach* is generally not as good as the approach of *retrieval by motion*. This may due to the use of color feature. As indicated in Table VII, color feature alone is not as discriminatory as motion feature. Even when color feature is combined with motion feature, the performance is still worse compared to the use of motion feature alone. Integration of multiple features for effective retrieval is still an open research issue. Our proposed work shows that when color and motion are combined in a hierarchical manner, certain degree of improvement is attained.

2) Soccer Video: We refer to the two soccer teams as teams A and B respectively. The color of the audiences' clothing is same as the players whom they support. Fig. 7 depicts the clustering results by using L_1 norm as the distance measure. Similarly, shots that are the nearest to cluster centroids are shown in the figure to represent clusters. The top level has six clusters, while the bottom level consists of eleven subclusters. By manual investigation of the clustering results, we summarize the characteristics of each cluster as follows.

- Cluster *c.1* mostly consists of players and audiences of team A, coaches of both teams, referees, and shots of players being hurt accidentally. The audiences of team A are all clustered in the subcluster *m.1.2*. Meanwhile, the player tracking shots are included in the subcluster *m.1.1*.
- Cluster *c.2* basically consists of players from both teams, with more players from team B. Nevertheless, the centroid of the subclusters *m.2.1* and *m.2.2* are the players from team A. This is due to the fact that the background color of these two shots is similar to the players' clothing in team B. This directly implies the need for foreground and background segmentation. In cluster *c.2*, the audience of team B are all clustered in *m.2.1*. This subcluster mainly consists of snapshots of players with slight motion. Meanwhile, the subcluster *m.2.2* comprises players being tracked in the soccer field.
- Cluster *c.3* has only two shots which are shown prior to the start of the soccer game.

TABLE VII
MEAN PRECISION OF DIFFERENT RETRIEVAL APPROACHES FOR BASKETBALL VIDEO

Approach	Mean Precision	
	L_1 norm	L_2 norm
Cluster-based Retrieval	0.532*	0.526
Retrieval by motion and color	0.510	0.426
Retrieval by motion	0.530	0.530
Retrieval by color	0.432	0.417

*The mark * indicates the best performance.*

- Cluster *c.4* has mainly the bird view of the soccer game. In *m.4.1*, the camera motion is stationary; in *m.4.2*, the camera pans to the left and to the right when one team attacks the other.
- Cluster *c.5* has mostly the medium shots of players passing the ball around. The camera motion in *m.5.1* is stationary, while the camera motion in *m.5.2* tracks the players when ditching the ball and facing opponents.
- Cluster *c.6* has one shot screening the sky. This shot is different from others in terms of color and content.

To test the retrieval performance, a query set which consists of 53 queries are manually picked and checked to have good answers. They consist of close up of players from different teams, bird view and medium shots, audience from different teams, and shooting shots. Players and audience from different teams are placed in different classes. Similarly, shots with different camera motions are put in different classes. Thus, the effectiveness of discriminating shots by color and motion can be experimented.

Fig. 8(a) shows the recall-precision of the tested approaches by using L_1 norm as distance measure, while Fig. 8(b) shows the recall-precision by using L_2 norm as distance measure. Table VIII summarizes the 11-point average precision values of various approaches. Similar to the experiment on the basketball video, L_1 norm is superior to L_2 norm in term of mean precision. The main results based on L_1 norm are: retrieval by both color and motion features is constantly superior to retrieval by either one feature; the recall of cluster-based retrieval is better than that of cluster-free retrieval.

3) Speed Efficiency: Table IX compares the motion and color features in term of the feature extraction time per image frame, and the feature vector length. The DC image size is 30×44. For the basketball video (122 shots), the clustering algorithm takes about 45 second to form a two-level hierarchical structure, while for the soccer video (404 shots), the algorithm takes approximately 380 sec (6.34 min). Table X further shows the average retrieval speed of 400 queries by the four tested approaches in the soccer video database. Cluster-based retrieval approach is about two times faster than of cluster-free approach (retrieval by motion and color features).

E. Discussion

We have presented the proposed two-level hierarchical clustering algorithm, together with the cluster-based vs. cluster-free

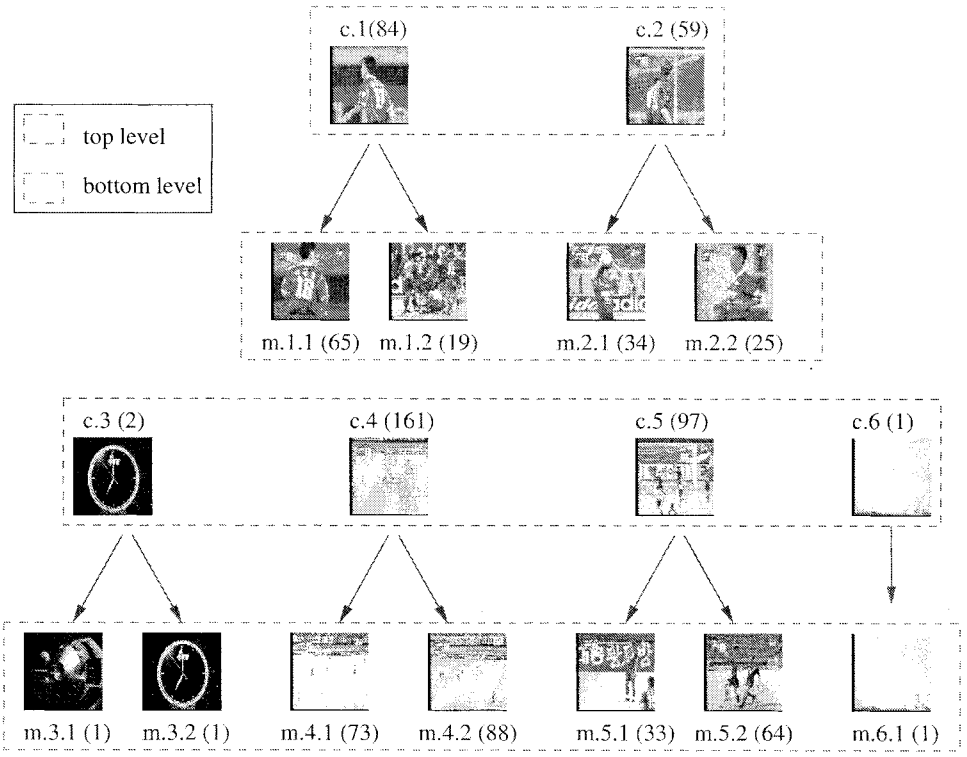
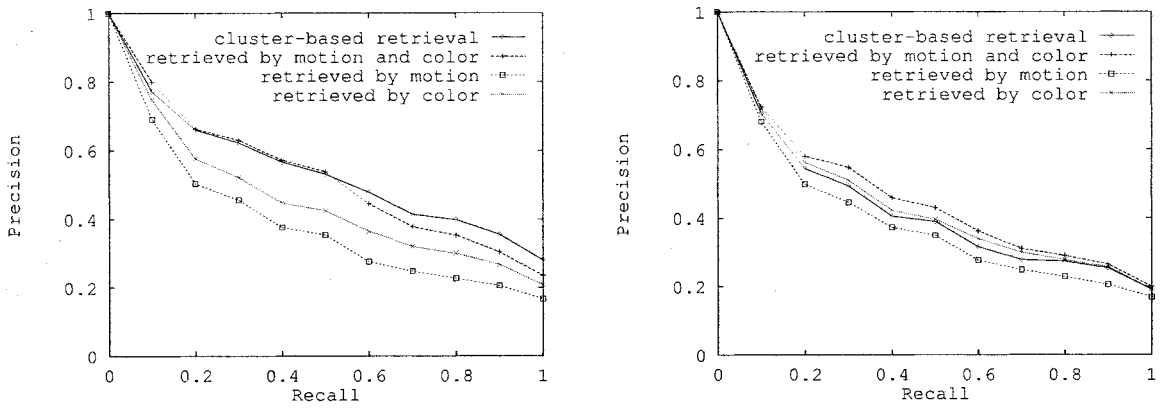


Fig. 7. Clustering result of soccer video by using L_1 norm as distance measure. $X(Y)$: X is cluster label and Y is the number of shots in a cluster.



(a) (b)

Fig. 8. Recall and precision curves for soccer video. (a) L_1 norm distance measure and (b) L_2 norm distance measure.

TABLE VIII
MEAN PRECISION OF DIFFERENT RETRIEVAL APPROACHES FOR SOCCER VIDEO

Approach	Mean Precision	
	L_1 norm	L_2 norm
Cluster-based Retrieval	0.553*	0.441
Retrieval by motion and color	0.538	0.470
Retrieval by motion	0.409	0.407
Retrieval by color	0.471	0.450

*The mark * indicates the best performance.*

TABLE IX
PERFORMANCE OF MOTION AND COLOR FEATURES (ON A PENTIUM III PLATFORM)

	Motion	Color
Feature extraction (sec)	0.028	0.0054
Feature vector length	8	64

retrieval methods. Through experiments, the clustering algorithm can successfully classify the content of basketball and soccer videos. Nevertheless, it is expected that player segmentation and hand-crafted domain specific knowledge will further improve the classification results. For retrieval, cluster-based approach in general gives slightly better results than that of cluster-free approach.

Currently, the precision of cluster validity analysis is assessed indirectly through the performance effectiveness of

TABLE X
RETRIEVAL SPEED ON A DATABASE OF 404 SHOTS (ON A SUN SPARC ULTRA-1 MACHINE)

Approach	speed (sec per shot)
Cluster-based Retrieval	0.144
Retrieval by motion and color	0.433
Retrieval by motion	0.146
Retrieval by color	0.310

cluster-based retrieval.[3] We would expect satisfactory retrieval accuracy if the precision is high. Nevertheless, if some error exists in clustering, currently there is no mechanism to get better retrieval results. However, it is possible to correct the error through relevancy feedback (RF) mechanism [21]. The issues of updating cluster structure through RF will be considered by us in future.

V. CONCLUSION

We have described the issues of clustering and retrieval for video abstraction and browsing. We start by proposing various methods to extract texture features from temporal slices for motion retrieval. Since tensor histogram features offer the best performance, we further combine tensor histograms and color histograms for clustering and retrieving of video shots. The validity of the proposed approaches have been confirmed by extensive and rigorous experimentations in sport video domain.

To apply the proposed methods to other video sources such as movie and documentary films, the current works need to be further pursued in two directions: motion segmentation and the integration of various video features. Unlike sport videos, the camera motion of most video sources is unrestricted. In most cases, the camera motion reveals what to be seen, but not the different ways of seeing events (which in turn tells the type of events) as what happens in sport videos. Therefore, the decomposition of camera and object motion is a preferable step prior to feature extraction. For the feature integration, the proposed motion and color features can be easily incorporated with other features such as audio and textual information for a more sophisticated video retrieval system.

APPENDIX A
RECALL-PRECISION (RP)

RP is computed as

$$recall = \frac{\text{number of relevant shots retrieved}}{\text{total number of relevant shots in database}}$$

$$precision = \frac{\text{number of relevant shots retrieved}}{\text{total number of shots retrieved}}.$$

Recall measures the ability to present all relevant items, while *precision* measures the ability to present only relevant items. *Recall* and *precision* are in the interval of [0, 1]. The recall-precision curve indicates a system's ability in ranking the relevant items. Ideally, precision values should be equal to one across all recall values.

[3]It is difficult to access directly the precision of cluster validity analysis since some shots do not belongs to any class that are interested to us (e.g., we do no categorize shots with closeup of referees and shots with players being hurt accidentally).

APPENDIX B
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

$$\text{ANMRR} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{\text{MRR}(q)}{C(q)+0.5-0.5\times NR(q)} \tag{22}$$

where

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

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

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

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

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 of rank lists.

REFERENCES

[1] E. H. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer.*, vol. 2, no. 2, pp. 284–299, Feb. 1985.

[2] R. C. Bolles and H. H. Baker, "Epipolar plane image analysis: An approach to determining structure from motion," *Int. J. Comput. Vis.*, vol. 1, no. 1, pp. 7–55, 1987.

[3] P. Bouthemy and R. Fablet, "Motion characterization from temporal cooccurrences of local motion-based measures for video indexing," in *Int. Conf. Pattern Recognition*, 1998, pp. 905–908.

[4] A. C. Bovik, M. Clark, and E. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, Jan. 1990.

[5] S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automatic content-based video search engine supporting multi-object spatio–temporal queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 602–615, Sept. 1998.

[6] R. Fablet, P. Bouthemy, and P. Perez, "Statistical motion-based video indexing and retrieval," in *Int. Conf. on Content-Based Multimedia Info. Access*, 2000, pp. 602–619.

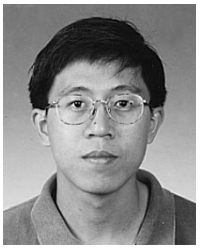
[7] G. H. Granlund and H. Knutsson, *Signal Processing for Computer Vision*. Dordrecht, The Netherlands: Kluwer, 1995.

[8] A. Hanjalic and H. J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1280–1289, Dec. 1999.

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, Nov. 1973.

[10] D. J. Heeger, "Model for the extraction of image flow," *Journal of Optical Society of America*, vol. 4, no. 8, pp. 1987–1471, Aug. 1987.

[11] G. Iyengar and A. B. Lipman, "Models for automatic classification of video sequences," *Proc. SPIE, Storage and Retrieval for Image and Video Databases VI*, pp. 3312–3334, 1998.
[12] A. K. Jain, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[13] B. Jähne, *Spatio–Temporal Image Processing: Theory and Scientific Applications*. Berlin, Germany: Springer Verlag, 1991.
[14] F. Liu and R. Picard, "Finding periodicity in space and time," in *Int. Conf. Computer Vision*, 1998, pp. 376–383.
[15] B. S. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.
[16] *MPEG-7 Visual Part of eXperimentation Model (XM) Version 2.0*, Dec. 1999.
[17] R. Nelson and R. Polana, "Qualitative recognition of motion using temporal texture," *Comput., Vis., Graph., Image Process.: Image Understand.*, vol. 56, no. 1, pp. 78–99, July 1992.
[18] C. W. Ngo, T. C. Pong, and R. T. Chin, "Detection of gradual transitions through temporal slice analysis," *Comput. Vis. Pattern Recognit.*, vol. 1, pp. 36–41, 1999.
[19] C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin, "Motion characterization by temporal slice analysis," *Comput. Vis. Pattern Recognit.*, vol. 2, pp. 768–773, 2000.
[20] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.
[21] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture for content-based multimedia information retrieval systems," in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997, pp. 82–89.
[22] E. Sahouria and A. Zakhor, "Content analysis of video using principle components," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1290–1298, Dec. 1999.
[23] E. P. Simoncelli, "Distributed representation and analysis of visual motion," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1993.
[24] M. O. Szummer, "Temporal texture modeling," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1995.
[25] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Trans. Circuits Syst. Video Technology*, vol. 10, pp. 133–146, Feb. 2000.
[26] A. B. Watson and A. J. Ahumada, "Model of human visual-motion sensing," *J. Opt. Soc. Amer.*, vol. 2, no. 2, pp. 322–341, Feb. 1985.
[27] B. L. Yeo and B. Liu, "On the extraction of dc sequence from mpeg compressed video," in *IEEE Int. Conf. Image Processing*, vol. 2, 1995, pp. 260–263.



Chong-Wah Ngo (M'02) received the B.S. degree (with honors) in 1994 and the M.S. degree in 1996, both in computer engineering, from Nanyang Technological University, Singapore, and the Ph.D. degree from the Hong Kong University of Science & Technology (HKUST) in 2000.

Since 2002, he has been an Assistant Professor with the City University of Hong Kong (CityU). Before joining CityU, he was a post-doctoral visitor at the Beckman Institute, University of Illinois at Urbana-Champaign and was a Research Associate at HKUST. He was with the Information Technology Institute, Singapore, in 1996, and was with Microsoft Research China, Beijing, as a summer intern in 1999. His current research interests include image & video indexing, computer vision and pattern recognition.

Ting-Chuen Pong received the Ph.D. degree in computer science from Virginia Polytechnic Institute and State University, Blacksburg, in 1984.

In 1991, he joined the Hong Kong University of Science & Technology, where he is currently a Reader of computer science and Associate Dean of Engineering. Before joining HKUST, he was an Associate Professor in computer science at the University of Minnesota, Minneapolis. He is currently on the Editorial Board of *Pattern Recognition*.

Dr. Pong was a recipient of the Annual Pattern Recognition Society Award in 1990 and Honorable Mention Award in 1986. He has served as Program Co-Chair of the Third International Computer Science Conference in 1995 and the Third Asian Conference on Computer Vision in 1998.

Hong-Jiang Zhang (S'90–M'91–SM'97) received the B.S. degree from Zheng Zhou University, China, and the Ph.D. degree from the Technical University of Denmark, Lyngby, both in electrical engineering, in 1982 and 1991, respectively.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at the MIT Media Lab, Massachusetts Institute of Technology, Cambridge, in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management; intelligent image processing and Internet media. In 1999, he joined Microsoft Research China, Beijing, where he is currently a Senior Researcher and the Assistant Managing Director, mainly in charge of media computing and information processing research. He has authored three books, over 120 refereed papers and book chapters, seven special issues of international journals in multimedia processing, content-based media retrieval, and Internet media, and has numerous patents or pending applications.

Dr. Zhang is a member of ACM. He currently serves on the editorial boards of five professional journals and a dozen committees of international conferences.