Local Features Based Image Sequence Retrieval

Xiang Fu and Jie-xian Zeng
Nanchang HangKong University, Nanchang, China
Email: fxfb163@163.com

***Abstract*—We propose an approach to retrieve image sequences similar to the given query image sequence from database. Our proposed approach consists of three phases. First, the query image sequence and every database image sequence are segmented into several shots based on histogram difference between local images of consecutive frames. Second, for each shot, one or more key frames are selected based on histogram difference between local images of benchmark frame and each followed frame. Third, to retrieve image sequences similar to the given query image sequence, similarity between the key frames of query image sequence shot and key frames of each database image sequence is computed. The similarity is also measured using the histogram difference between local images of key frames. Local image is defined by interest points, that is, the regions around all interest points for a frame. The database video shots with similarity higher than a predefined threshold are output and returned to the user. Experimental results show that the proposed video shot detection method can overcome the deficiency of traditional histogram-based method that different contents frames have similar histograms, can distinguish between gradual changes and camera motions effectively, and can effectively detect both abrupt transitions and gradual transitions; the key frames selected by the proposed method have good representative power and can improve the performance of video shot retrieval.**

***Index Terms*—image sequence retrieval; shot boundary detection; key frame extraction; local image; interest points**

I. INTRODUCTION

Today's advanced digital media technology has let to the explosive growth of multimedia data in a scale that has never occurred before. The availability of such large-scale quantities of multimedia documents prompts the need for efficient algorithms to search and index multimedia files. Video clips are an important type of multimedia data, indexing and retrieval of digital video is a very active research area.

Previous work on video retrieval can be classified into two main streams: keywords-based methods and key frames-based methods. For automatic video retrieval, it is almost impossible to use keywords to describe video sequences [1]. The reasons are that this process requires tremendous manpower, and the keywords to be used are subjective. A generic approach for managing video data is first to segment a video into groups of related frames called "shots" by means of shot detection. After identifying the shot boundaries, one or more key frames can be extracted for each video shot. The visual contents of these key frames are then used to represent the video shots for indexing and retrieval.

After comparing most general video shot retrieval algorithms, we have found that there are 3 main components that affect the performance of retrieval. These are

1. Video shot boundary detection is the first step of video indexing. Because of the unstructured property and content variety of video sequence, the automatic video shot partition is still a challenging research problem.

2. Key frame extraction method. The key frames should provide the most representative power in video shot representation. The followed video retrieval methods are implemented on key frames rather than each frame to reduce the computational expenses.

3. The feature matching is useful for finding contents similar to a query video stream from video database. The feature matching is the process of video indexing actually.

In this paper, we research from these three aspects, new methods based on local features are proposed. Firstly, the shot boundary detection method is based on the histogram difference between local images of consecutive frames. The local images are defined by interest points, that is, the regions around all interest points. Thus it is more sensitive to the difference between frames than traditional histogram-based shot boundary detection method. So it can overcome the deficiency of traditional histogram-based method that different contents frames have similar histograms, and can distinguish between gradual changes and camera motions effectively. Secondly, the key frames extraction method is based on the histogram difference between local images of the benchmark frame and each followed frame for a video shot. The method is computationally effective and the key frames have good representative power in video shot representation. Finally, video shot indexing is implemented based on the histogram difference between local images of the key frames for a query video shot and key frames for shots of video database.

The remainder of the paper is organized as follows: The description of the proposed shot detection method,

This work was supported partially by the National Natural Science Foundation of China (Grant No. 60675022), the Natural Science Foundation of JiangXi, China (Grant No. 2008GZS0034) and Aviation Science Foundation of China (Grant No. 20085556017).

Corresponding author: Xiang Fu

doi:10.4304/jcp.5.7.987-994

key frames extraction method and video indexing method are addressed in Section II, III and IV respectively. Experimental results are presented and commented in Section V and conclusions are drawn in Section VI.

II. SHOT BOUNDARY DETECTION

Shot boundary detection is the first step towards further analysis of the video content for indexing, shot classification, browsing, searching and summarization. To form a video sequence, shots are joined together during video sorting or post editing with either abrupt cuts or gradual visual effects such as dissolves, wipes, and fades. Normally different types of shot transitions are selected deliberately by the editor to indicate different events. For example, fades normally correspond to the start or end of a program.

During the pass two decades, quite a lot of approaches to shot boundary detection were proposed. There are broadly two approaches, namely, the pixel domain processing and the compressed domain processing. In the first approach, frame pixels are processed after decoding a compressed video file like MPEG. Pixel domain methods include pixel-to-pixel frame comparison, block-to-block frame comparison and histogram-based frame comparison [2]. In contrast to these methods, compressed domain approaches make direct use of the encoded data. Lee et al exploit information from the first few AC coefficients in the transformation domain, and track binary edge maps to segment the video [3]. Statistical sequential analysis technique is proposed to detect scene change on the DC image sequence [4]. Macroblock-based methods also work on compressed MEPG digital video [5]. Zeng and Gao proposed a set of methods to deal with shot-change detection problem on H.264/AVC compressed video [6].

Comparison of various shot detection methods has shown that the pixel domain methods have higher accuracy compared to the compressed domain methods [7,8]. The compressed domain methods, on the other hand, work faster. Also, of all the pixel domain methods, histogram based techniques perform better than the rest [2]. However, for the histogram-based methods, it is possible that the histograms of two frames are similar, but the contents are completely different [9]. In order to handle this situation, we propose a video shot detection method using local color features. The method can distinguish between gradual changes and camera motions effectively, and can detect both abrupt transitions and gradual transitions effectively.

The basic idea of our shot boundary detection method is computing the histogram of local images based on interest point. As mentioned above, for histogram based shot boundary detection techniques perform better than the rest, our method is also histogram based. For frames with different contents have different interest points, which define different local images with different histogram. So the proposed method can overcome the deficiency of traditional histogram based method that different contents frames have similar histograms. The proposed method can reduce the effects of object motions in the scene or camera motion for similar reason.

The flowchart of the proposed shot detection algorithm is shown in Fig.1. It contains four main steps: *Interest Point Detection*, *Local Image Extraction*, *Histogram of Local Image*, and *Shot Boundary Detection*.

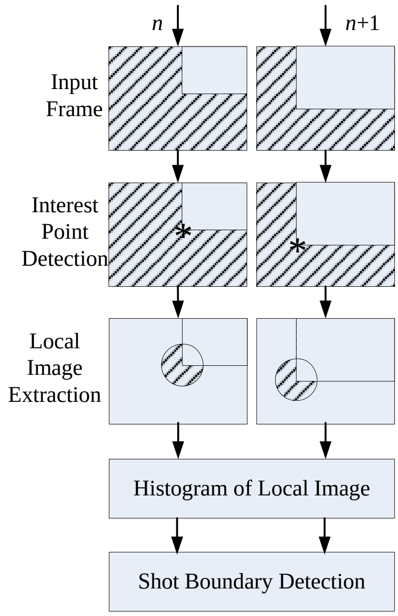


Figure 1. Flowchart of shot boundary detection

A Interest Point Detection

Many different interest point detectors exist in the literature. The Harris detector proposed by Harris and Stephens [10] is widely used for its good performance and invariance to rotation and translation. The idea of Harris interest point detector is to detect locations in a frame f where the pixel values have significant variations in both directions. For a given scale of observation σ_l^2, such interest points can be found from a windowed second moment matrix integrated at scale $\sigma_i^2 = s\sigma_l^2$:

$$\mu = g(\cdot;\sigma_i^2) * \begin{pmatrix} (L_x)^2 & L_x L_y \\ L_x L_y & (L_y)^2 \end{pmatrix} \quad (1)$$

where L_x and L_y are Gaussian derivatives defined as:

$$\begin{cases} L_x(\cdot;\sigma_l^2) = \partial_x (g(\cdot;\sigma_l^2) * f) \\ L_y(\cdot;\sigma_l^2) = \partial_y (g(\cdot;\sigma_l^2) * f) \end{cases} \quad (2)$$

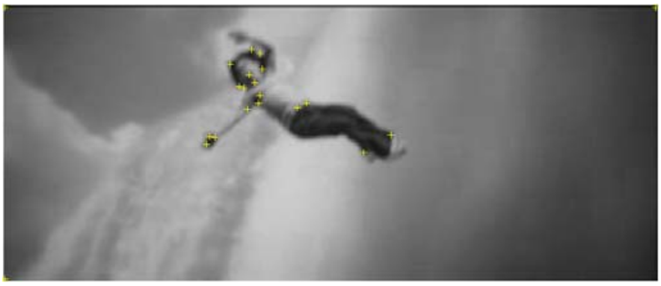
and where g is the Gaussian kernel:

$$g(x, y;\sigma^2) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/2\sigma^2) \quad (3)$$

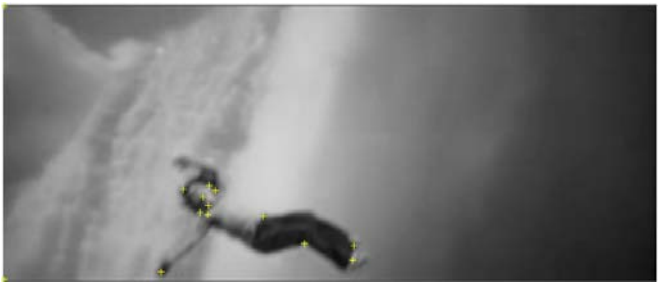
As the eigenvalues $\lambda_1, \lambda_2, (\lambda_1 \le \lambda_2)$ of μ represent characteristic variations of f in both image directions, two significant values of λ_1, λ_2 indicate the presence of and interest point. To detect such points, Harris and Stephens proposed to detect positive maxima of the corner function:

$$H = \det(\mu) - k \cdot trace^2(\mu) = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (4)$$

Fig.2 shows examples of interest points detected by Harris detector. From Fig.2(a), it can be seen that the interest points gathered in the regions of clearer foreground objects. Comparing Fig.2(a) with Fig.2(b), the background has changed for the camera moving with objects, while the interest points can keep stable.



(a)



(b)

Figure 2. Examples of interest points detected by Harris detector. The background has changed for the camera moving with objects.

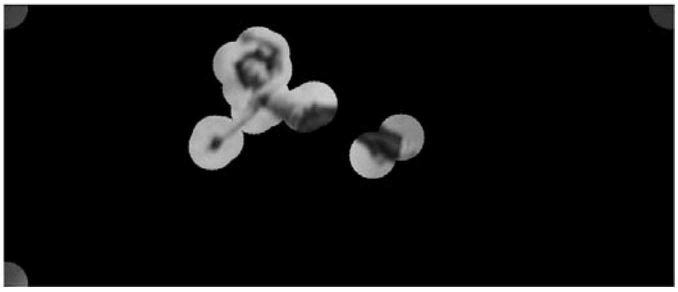
B Local Image Extraction

The local image is consisting of regions around interest points extracted in the previous step. The radius is set to R pixels, centered on the interest point, as shown in Fig.3, where the value of R is 30 pixels. The local regions around interest points were used in the following computing, while the black areas which corresponding to stable regions were ignored. The value of R is variable. A smaller value of R means a larger stable area will be ignored and the more of the local region be emphasized, thus the more sensitive to the difference between frames.

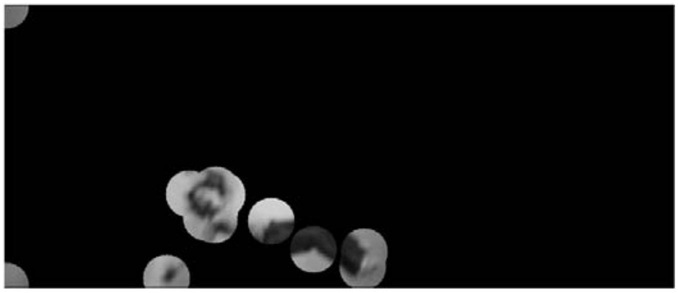
C Histogram of Local Image

For shot boundary detecting, we compute the histogram of local image obtained in last step. The traditional histogram is easy to compute, and performs generally better than other features when used to detect shot boundary, but it is insensitive to small local object motion or partial occlusion. Furthermore, it is possible that the histograms of two frames are similar, but the contents are completely different. As shown in Fig.4, Fig.4(a) and Fig.4(b) are artificial images, the values on each block is its gray value, the largest block in the middle of Fig4(a) has value of 110 is interpolated into other blocks in Fig.4(b). Undoubtedly, they have same histograms as shown in Fig.4(e).

In our algorithm, the gray histogram of local image is computed. In Fig.4(a) and Fig.4(b), the interest points are also described as little cross, the corresponding local images are Fig.4(c) and Fig.4(d). And the histograms of two local images are shown in Fig.4(f), it can be seen that the histograms become different completely. Especially, the bin of histogram of gray value 110 for Fig.4(d) is larger than that of Fig.4(c), where a larger smooth area was ignored.



(a)



(b)

Figure 3. Local images defined by interest points in Fig.2.

D Shot Boundary Detection

The shot boundary detection principle is very important for a shot detection algorithm. And gradual transitions are generally more difficult to be detected than abrupt cuts. It is found that for a shot cut, the change between consecutive frames is large. However, for gradual transitions, the change between consecutive frames is not large enough, but the number of change between the frames before and after the transition is large. These are the general temporal properties of shot cuts and gradual transitions.

The change between frames is measured by histogram difference. The histogram difference of frame F_i and frame F_{i-1} is:

$$d(H_i, H_{i-1}) = \sum_{k=0}^{N-1} \left\| H_i(k) - H_{i-1}(k) \right\|^q \quad (5)$$

where N is the number of histogram bins, and q the order of norm distance which is set to 1 for its good performance and easy computation.

When perform shot boundary detection, we not use the histogram difference directly, we use the difference of histogram difference instead.

The difference of histogram difference is:

$$Dd = d(H_i, H_{i-1}) - d(H_{i-1}, H_{i-2}) \tag{6}$$

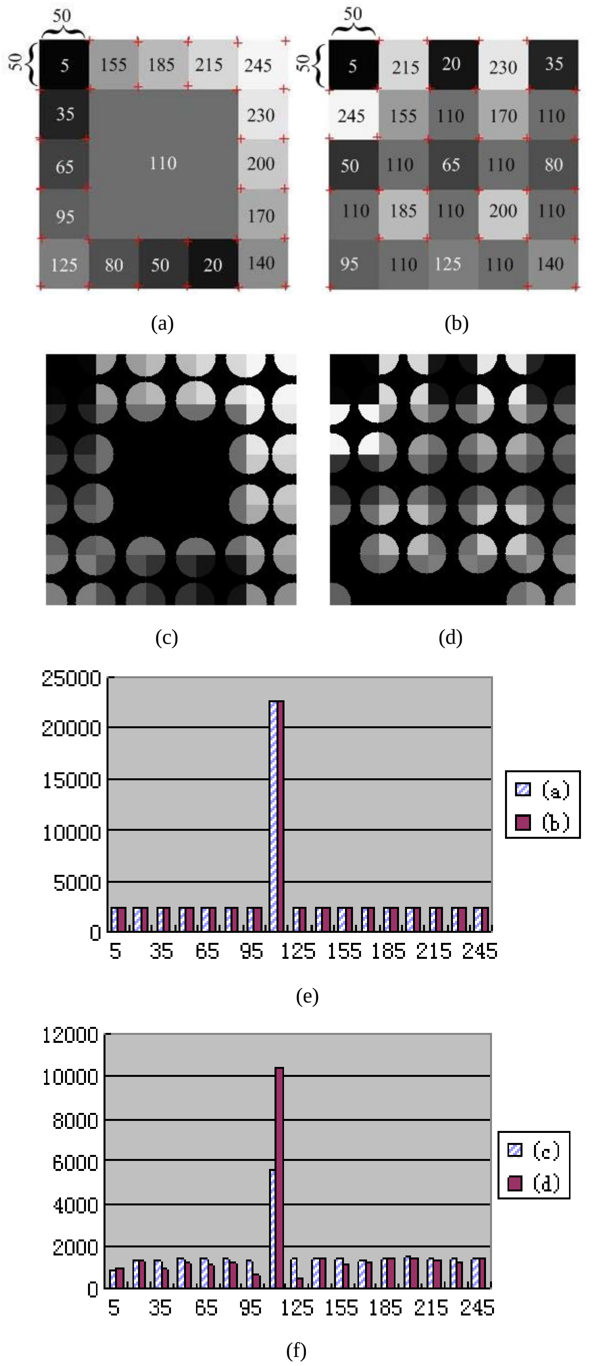
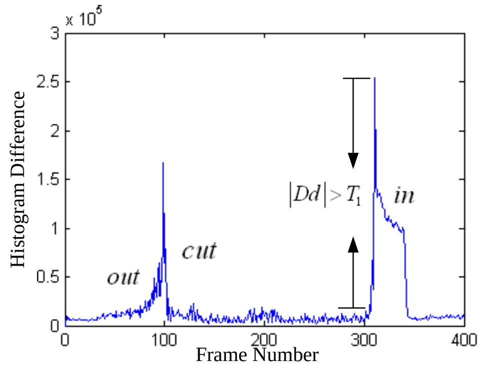


Figure 4. Histogram of original images and local images. (a) and (b) have the same gray distribution, and have the same histograms as shown in (e). Local images of (a) and (b) are shown in (c) and (d) respectively, they have different histograms as described in (f).

The value of Dd indicates the change tendency of consecutive frames. $Dd > 0$ means the gray of frames changed more and more great, and $Dd < 0$ means more and more slow. Shot boundary can be detected as follows:

$$\begin{cases} |Dd| > T_1 & & cut \\ |Dd| < T_1 \quad and \quad N_{Dd>0} > T_2 & & out \\ |Dd| < T_1 \quad and \quad N_{Dd<0} > T_2 & & in \end{cases} \tag{7}$$

where T_1 is the cut threshold, $|Dd| > T_1$ means a great change is happened and a abrupt cut is detected; $N_{Dd>0}$ is the number of frames that $Dd > 0$, $N_{Dd<0}$ is the number of frames that $Dd < 0$, and T_2 is the gradual transition threshold. $|Dd| < T_1$ and $N_{Dd>0} > T_2$ means the current shot is going out slowly, and $|Dd| < T_1$ and $N_{Dd<0} > T_2$ means next shot is coming in slowly, these were described in Fig.5.



(*Dd*: Difference of Histogram difference)

Figure 5. Video shot boundaries are detected based on the difference of histogram difference.

III. Key Frame Extraction

Key frames representation is a simple yet effective way of summarizing the content of videos for browsing and retrieval. The use of key frames greatly reduces the amount of data required in video indexing and browsing with video content. Key frame extraction is the process of properly selecting frames directly from shots to represent the video content. The main challenge lies in how to represent videos with key frames in a perceptually compact and meaningful way.

Some simple methods pick one or more key frames from every shot at predetermined temporal locations such as the first, middle and/or the last frame [11,12]. In [13] frames of a video sequence are chosen at regular time intervals, leading to a storyboard presentation. Although these methods are simple and computationally efficient, they may not provide the most representative power in video shot representation, especially for shots of long duration and high motion activity. Better key frames could be chosen if shot content were considered in the selection.

It is found that the common problems of key frames extraction techniques are that advanced schemes have high complexity, while simpler schemes do not provide enough visual information of each shot. To overcome these drawbacks, in this section, an interest points based key frames extraction method is presented.

A given sequence is segmented into video shots using the shot detection method in Section II. For a video shot, the first frame is chosen as a key frame. The first frame is treated as a benchmark frame, for all the other frames of the shot, executing the following ten steps:

Step 1: Detecting the interest points of benchmark frame using (1)~(4);

Step 2: Extracting the local image of benchmark frame based on the interest points;

Step 3: Computing the histogram of local image of benchmark frame;

Step 4: Detecting the interest points of the $i-th$ frame using (1)~(4);

Step 5: Extracting the local images of the $i-th$ frame based on the interest points;

Step 6: Computing the histogram of local images of the $i-th$ frame;

Step 7: Computing the histogram difference $d(H_B, H_i)$ between the benchmark frame and the $i-th$ frame using (5). Where H_B is the histogram of local image of the benchmark frame and H_i is the histogram of local image of the $i-th$ frame.

Step 8: If $d(H_B, H_i) > T_3$, the $i-th$ frame is selected as key frame and new benchmark frame, T_3 is the threshold for key frames choosing.

Step 9: Let $i = i+1$, that is, the next frame will be processed in next step;

Step 10: Return back to Step 1 if new benchmark frame is selected in Step 8, otherwise return back to Step 4. Repeat above steps until all the frames of a video shot been processed.

The process of key frame selection can be described as Fig.6. Firstly, the first frame of a video shot is selected as key frame and benchmark frame, and the followed frames compared with the benchmark frame one by one, for the difference between frame 1 and frame k is large enough, frame k is selected as new key frame and new benchmark frame. Then the followed frames compared with the new benchmark frame one by one. Repeat the process until all the frames of a video shot processed, and different numbers of key frames are extracted for video shots of a sequence.

The key frames extraction method is based on the histogram difference of local images between the benchmark frame and each followed frame for a video shot, so the key frames have good representative power in video shot representation. The method is processed based on interest points, which can reduce the amount of data greatly, what's more, the histograms of local images for all frames have been obtained at the stage of shot boundary detection in Section II and can be used directly here, so the method is computationally effective.

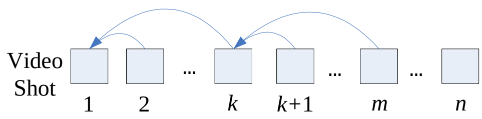


Figure 6. Process of key frame selection for a video shot

IV. VIDEO INDEXING

Video indexing is the process of features matching of key frames between the query video shot and the video shots of database. So the query video sequence and the video sequences of database should be segmented into shots using the method in Section II firstly, and then key frames should be extracted using the method in Section III. These steps can be considered as the preprocessing for video indexing.

There are many methods for features matching have been processed in literatures. The video indexing base on key frames is similar to the static image retrieval, so the traditional technologies of image retrieval can be used here. Here we also use the histogram difference of local images as features for video indexing, as following nine steps:

Step 1: For key frame i of query video shot, detecting the interest points using (1)~(4);

Step 2: Extracting the local image of frame i based on the interest points;

Step 3: Computing the histogram of local image of frame i;

Step 4: For key frame j of database video shot k, detecting the interest points using (1)~(4);

Step 5: Extracting the local images of frame j based on the interest points;

Step 6: Computing the histogram of local images of frame j;

Step 7: Computing the histogram difference $d(H_i, H_{kj})$ between the key frames of query shot and database shot using (5). Where H_i is the local image histogram of key frame i for query video shot and H_{kj} is the local image histogram of key frame j for database video shot k.

Step 8: If $d(H_i, H_{kj}) < T_4$, which means the difference between two frames is small enough, database video shot k is outputted as the retrieval result, and the whole process is finished. Where T_4 is the threshold for key frames matching.

Step 9: If $d(H_i, H_{kj}) > T_4$, let $j = j+1$, that is, the next key frame of database video shot k will be processed in next step; If all the key frames of database video shot k have been processed, let $k = k+1$, which means the next database video shot will be processed in next step; If all the database video shots have been processed, let $i = i+1$, which means the next key frame of the query video shot will be used; If all the key frames of the query video shot have been used, then output N

database video shots with the minimum $d(H_i, H_{kj})$ as the retrieval results. As shown in Fig.7, all the key frames of query video shot will be compared with all the key frames of database video shots if there is no satisfactory result obtained.

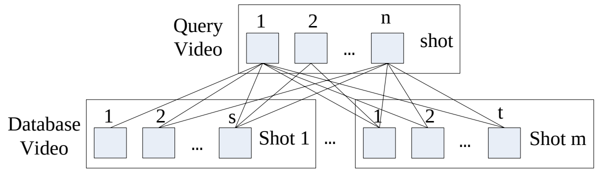


Figure 7. Process of key frames matching for video indexing

V. EXPERIMENTAL RESULTS

A Video shot boundary detection

To evaluate the performance of the proposed shot detection algorithm, we use various kinds of videos from news reports and famous movies, parts of them as shown in Table I. The sequence of Earthquake Report has lots of gradual changes; Superman and Tomb Raider have many fighting scenes and the camera moving greatly; and the Friends has a lot of abrupt cuts accompanied with captions fade in and fade out.

The comparison between the proposed algorithm and traditional Histogram method relies on the recall precision and false alarms rate

$$\text{recall} = \text{detects}/(\text{detects} + MD) \quad (8)$$

$$\text{false alarms} = FA/(\text{detects} + MD) \quad (9)$$

where detects denotes the correctly detected boundaries, while MD and FA denote missed detections and false alarms. Results of recall for abrupt cuts and gradual changes detection are provided in Tables II and III, respectively. It can be seen that the performance of the traditional Histogram method is close to the proposed algorithm for abrupt cuts detection, while the proposed algorithm is outperformed to the former for gradual changes detection. That's because the gradual changed frames have similar histograms, but have different interest points. So the local images defined based on the interest points are more sensitive to gradual changes between frames.

The large movement of camera will lead to false detection for Histogram-based method, while the proposed method is suitable for this situation, because when the camera moving with objects, most of interest points keep homologies in the regions of objects. So the false alarms rates for Histogram are much higher than that of the proposed method, as shown in Table IV.

B Key frames extraction

There are no objective standards to measure the performance of key frames extraction method. In this section, some key frame extraction results of our method are shown in Fig.8. In the next section, experimental results will be given to see how the proposed method can improve the performance of video shot retrieval.

The hall_monitor sequence has static background, and two people come in and go out the scene slowly. The whole sequence is considered as a video shot by video shot detection method. And the key frames extracted by the proposed method are shown in Fig.8. It can be seen that key frames have good representative power in video shot representation.

TABLE I. DESCRIPTION OF THE VIDEO SEQUENCES IN THE TEST SET

Video	No. of Frames	No. of abrupt Cuts	No. of gradual changes
Earthquake Report	2000	6	9
Superman (Movie)	755	3	0
Friends (Movie)	2000	73	8
Tomb Raider (Movie)	960	31	0
(*Total*)	5715	113	17

TABLE II. ABRUPT TRANSITION PERFORMANCE

Video	Histogram	Proposed
Earthquake Report	5	6
Superman (Movie)	3	3
Friends (Movie)	64	70
Tomb Raider (Movie)	28	25
	88.50%	92.04%

TABLE III. GRADUAL PERFORMANCE

Video	Histogram	Proposed
Earthquake Report	5	7
Superman (Movie)	0	0
Friends (Movie)	5	7
Tomb Raider (Movie)	0	0
	58.82%	82.35%

TABLE IV. FALSE ALARMS

Video	Histogram	Proposed
Earthquake Report	7	7
Superman (Movie)	0	0
Friends (Movie)	15	6
Tomb Raider (Movie)	22	6
	33.85%	14.62%

C Video retrieval

To evaluate the performance of the video retrieval method, the Average Recall (AR) and Average

Normalized Modified Retrieval Rank (ANMRR) [1,14] are used.

$$AR = \frac{1}{Q}\sum_{q=1}^{Q}\frac{nr(q)}{ng(q)} \tag{10}$$

$$ANMRR = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) \tag{11}$$

$$NMRR(q) = \frac{MRR(q)}{K - \frac{ng(q)}{2} + 0.5} \tag{12}$$

$$MRR(q) = \sum_{i=1}^{ng(q)}\frac{r(i)}{ng(q)} - \frac{ng(q)}{2} - 0.5 \tag{13}$$

where $ng(q)$ is the number of similar videos with query video q, there are Q query videos totally; $nr(q)$ is the number of similar videos included in the first K retrieval results, $K = \min\{4 \times ng(q), 2 \times GTM\}$, $GTM = \max\{ng(q)\}$; $r(i) = Rank$ if similar videos are all included in the first K results, otherwise $r(i) = K + 1$.

Figure 8. Key frames selected for hall_monitor sequence.

These measures determine how many of the correct video shots are retrieved and how good their rankings are among the retrievals. In our video database, we collect 30 video sequences with about 1000 video shots. Some of these sequences are downloaded from the Web, including news reports, famous movies and sports records, and some of them are captured in our laboratory.

In the experiments, the proposed method is compared with the method TMOF in [1]. In TMOF, an optimal key frame is constructed based on global statistics. Each pixel in that optimal key frame is constructed by considering the probability of occurrence of those pixels at the corresponding pixel position among the frames in a video shot. The experimental results are illustrated in Table V. Experimental results show that our proposed method result in a lower ANMRR and a higher AR compared to TMOF. The reason is that more than one key frame are used to retrieval for the proposed method. The key frames have better representative power in video shot representation, and the video shots retrieval based on these key frames has better performance than TMOF.

TABLE V. PERFORMANCE OF DIFFERENT VIDEO RETRIEVAL METHOD

	ANMRR	AR
TMOF	0.3635	0.7176
Proposed	0.2644	0.9256

VI. CONCLUSIONS

In this paper, we present a new video retrieval approach based on local image defined by interest points. Our proposed approach consists of three phases. First, every video sequence is segmented into several shots based on histogram difference between local images of consecutive frames. Because frames with different contents have different interest points, which define different local images with different histogram, the proposed method can overcome the deficiency of traditional histogram-based method that different contents frames have similar histograms, and it can effectively detect gradual transitions for similar reasons. On the other hand, the large movement of camera will lead to false detection for Histogram-based method, while the proposed method is suitable for this situation, because when the camera moving with objects, most of interest points keep homologies in the regions of objects and the histogram of local images keep stable, thus it can distinguish between gradual transitions and camera motions effectively. Second, for each shot, one or more key frames are selected based on histogram difference between local images of benchmark frame and each followed frame. Third, video shot indexing is performed based on key frames. The local images' histogram difference between key frames of query video shot and database video shot is also computed. The database video shots with the difference lower than a predefined threshold are output and returned to the user. The key frames selected by the proposed method have good representative power and can improve the performance of video shot retrieval than traditional method.

We are not satisfied with the performance of the proposed approach for videos with moving and complicated background, such as football video with spectators as background. In these situations, the local image is ineffective for there are lots of interests points can be detected, especially in the background regions. In the future, these situations will be investigated, the idea of moving object tracking based on interest points will be included in our approach to improve the performance, and more sports videos with complicated moving background will be used to test our approach.

REFERENCES

[1] K. W. Sze, K. M. Lam and G. P. Qiu, "An optimal key frame representation for video shot retrieval," *In Proc. of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp.270-273,Oct.2004.

[2] A. Vadivel, M. Mohan, S. Sural and A. K. Majumda, "Object level frame comparison for video shot detection," *In Proc. of the IEEE Workshop on Motion and Video Computing*, pp.235-240, Jan. 2005.

[3] S. W. Lee, Y. M. Kim and S. W. Choi, Fast scene change detection using direct feature extraction from MPEG compressed videos," *IEEE Trans. Multimedia*, Vol.2, No.4, pp.240-254, Feb. 2000.

[4] D. Lelescu and D. Schonfeld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Trans. Multimedia*, Vol.5, No.1, pp.106-117, Mar. 2003.

[5] S. C. Pei and Y. Z. Chou, "Effective wipe detection in MPEG compressed video using macroblock type information," *IEEE Trans. Multimedia*, Vol.4, No.3, pp.309-319, Apr. 2002.

[6] W. Zeng and W. Gao, "Shot change detection on H.264/AVC compredssed video", *In International Symposium on Circuits and Systems*, pp.3459-3462, May. 2005.

[7] U. Gargi, R. Kasturi and S. H. Strayer, "Performance characterization of video shot-change detection methods," *IEEE Trans. Circuits and Systems for Video Technology*, Vol.CSVT-10, No.1, pp.1-13, 2000.

[8] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," *In Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, Vol.3656, pp.290-301, Jan. 1998.

[9] J. Zheng, F. M. Zou and M. Shi, "An efficient algorithm for video shot boundary detection," *In Proc. of International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp.266-269, Oct. 2004.

[10] C. harris and M. Stephens, "A combined corner and edge detector," *In Alvey Vision Conference*, pp.147-151, Sept. 1988.

[11] J. W. Rong, W. J. Jin and L. D. Wu, "Key frame extraction using inter-shot information," *In Proc. of International Conference on Multimedia and Expo*, pp.571-574, Jun. 2004.

[12] K. S. Ntalianis and S. D. Kollias, "An Optimized Key-Frames Extraction Scheme Based on SVD and Correlation Minimization," *IEEE Internatinal Conference on Multimedia and Expo*, pp.792-795, Jul. 2005.

[13] M. Mills, J. Cohen and Y. Y. Wong, "A magnifier tool for video data," *In Proc. ACM Computer Human Interface*, pp.93-98, May 1992.

[14] J. Feng, X. B. Gao and Y. M. Yang, "Key frame reconstruction algorithm for video shot retrieval," *Journal of Computer-Aided Design & Computer Graphics*, Vol.20, No.6, pp775-780., Jun.2008.

Xiang Fu was born He-feng,Hubei,China in 1980. He obtained a Master of Science (M.S.E.) degree in computer application and Ph.D degree in Circuits and Systems from Xidian University,Xi-an,China respectively in 2005 and 2008.

He is currently an instructor of Nanchang Hangkong University, Nanchang, China. His research interests include image processing and pattern recognition.

Jie-xian Zeng was born in Le-an,Jiangxi province,China in 1958.He obtained a Master of Science (M.S.E.) degree in engineering graphics from Northwestern Polytechnical University,Xi-an,China in 1997.

He is currently a professor of Nanchang Hangkong University China, and has published more than 50 papers. He has presided and achieved one National Science Foundtion of China under Grant(No: 60275037) and one National Science Foundtion of Jiangxi, China under Grant(No:0311019) respectively. His research interests include computer graphics, theoretical graphics, computer vision, and pattern recognition.