Within and Between Shot Information Utilisation in Video Key Frame Extraction

Dianting Liu^{*,‡}, Mei-Ling Shyu^{*,§}, Chao Chen^{*,¶} and Shu-Ching Chen^{†,||}

^{*}*Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33124, USA*

[†]*Distributed Multimedia Information Systems Laboratory
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA*

[‡]*d.liu4@umiami.edu*

[§]*shyu@miami.edu*

[¶]*c.chen15@umiami.edu*

^{||}*chens@cs.fiu.edu*

Abstract. In consequence of the popularity of family video recorders and the surge of Web 2.0, increasing amounts of videos have made the management and integration of the information in videos an urgent and important issue in video retrieval. Key frames, as a high-quality summary of videos, play an important role in the areas of video browsing, searching, categorisation, and indexing. An effective set of key frames should include major objects and events of the video sequence, and should contain minimum content redundancies. In this paper, an innovative key frame extraction method is proposed to select representative key frames for a video. By analysing the differences between frames and utilising the clustering technique, a set of key frame candidates (KFCs) is first selected at the shot level, and then the information within a video shot and between video shots is used to filter the candidate set to generate the final set of key frames. Experimental results on the TRECVID 2007 video dataset have demonstrated the effectiveness of our proposed key frame extraction method in terms of the percentage of the extracted key frames and the retrieval precision.

Keywords: Key frame extraction; clustering; information integration.

1. Introduction

As the cost of creating, acquiring, and transmitting videos sharply decreases, huge amounts of video data have been created and delivered every day. Millions of YouTube videos are clicked each day and meanwhile, hundreds of thousands of new videos are uploaded to the YouTube website. All of these create new demands on efficient video browsing, searching, categorisation, and indexing. Videos can be regarded as a sequence or combination of video frames which are basic units of a video. Usually, the amount of frames within a video is quite large. For example, a video that lasts for 10 minutes at a frame rate

of 25 frames per second has a total of 15,000 frames. The analysis of a video based on its frames could be computationally unaffordable if the video length is very long. Therefore, representative frames that are commonly called key frames are selected and extracted from a video. These extracted key frames are supposed to be able to describe the content of the video and summarise the contained information (Truong and Venkatesh, 2007; Fu *et al.*, 2010).

Today, some websites like MEGAVIDEO provide key frame-based browsing functionality to each video, so that a user who wants to briefly browse a video's content only needs to put the cursor on the interested video and glance at a sequence of key frames rather than to operate it in the traditional way — clicking and watching the whole video sequence. The key frame-based browsing functionality not only decreases the time that users spend in searching their favourite videos, but also reduces the network traffic load by delivering a few images rather than the whole video streaming. Key frames are also widely used in video searching and indexing tasks. Almost all the teams utilised key frames-based features from video sources in the TRECVID high-level feature extraction and semantic indexing task (Smeaton *et al.*, 2006). Key frames make it practical for each team to analyse video contents and construct learning/ranking models, providing a list of ranked shots by their relevance to the concerned high-level features or concepts.

Various categories of key frame extraction approaches have been developed, which used sampling, content changes, coverage-based, cluster-based, motion-based, colour-based, etc., techniques. Each category has its advantages and disadvantages. It is well-acknowledged

that the balance between the quality and the quantity of the extracted key frames is a major challenge that needs to be addressed.

In this paper, an innovative key frame extraction approach is proposed with an attempt to achieve a balance between the quantity of key frames for summarising the shots in a video sequence and the quality of key frames to represent the whole video. Our proposed approach has the following contributions: (1) it identifies transitive regions and informative regions by analysing the differences between consecutive frames at the shot level; (2) a modified clustering technique is utilised as the key frame extractor to select a set of key frame candidates (KFCs) in informative regions, while transitive regions are not used for key frame extraction; and (3) it integrates the frame information within a video shot and between video shots to filter redundant KFCs to generate the final set of key frames.

The rest of this paper is organised as follows. Section 2 discusses various categories of key frame extraction approaches. The proposed key frame extraction approach is presented in Section 3. In Section 4, the experimental results and analyses are provided followed by time complexity analysis. Finally, Section 5 concludes this paper and discusses the future work.

2. Related Work

The simplest way to get key frames is to use the sampling technique (Hoon *et al.*, 2000). For example, uniform sampling generates key frames at a fixed sampling rate. However, since the sampling methods do not consider the characteristics of non-homogeneous distribution of the video visual information, they suffer seriously from two major issues. First, the sampling results may miss a lot of important frames which contain the significant content of the videos. Second, since many frames within a shot are very visually similar to each other, content redundancies widely exist in the sampling results. To overcome these two problems, many key frame extraction algorithms have been proposed and developed.

One category of the algorithms generates key frames when the content change exceeds a certain threshold (Rasheed and Shah, 2003; Zhang *et al.*, 1997; Kang and Hua, 2005; Kim and Hwang, 2002; Zhang *et al.*, 2003). The content change could be measured by a function based on histograms, accumulated energy, etc. The algorithms belonging to this category do not need shot segmentation before applying key frame extraction methods. Therefore, they are suitable for real time applications. However, one problem for this category of algorithms is that key frames are generated without considering

the content of frames in the remaining video sequence. Therefore, the selected key frames may still contain lots of redundancies and become suboptimal, since they cannot represent the content temporally behind them. In other words, the content coverage of these key frames is only limited to the preceding frames.

To overcome the above problem, coverage-based approaches were proposed (Yahiaoui *et al.*, 2001; Rong *et al.*, 2004; Cooper and Foote, 2005; Chang *et al.*, 1999), which aim to get a small number of key frames by maximising each key frame's coverage towards adjacent frames. One method presented by Chang *et al.* (1999) applied greedy searching to find key frames with maximum coverage iteratively until all frames were represented by key frames. The major drawback of the coverage-based approaches is heavy computation. In order to search key frames according to coverage, dissimilarity scores need to be calculated on all pairs of frames. Therefore, the performance of coverage-based approaches is limited by the computation power of underlying hardware. Another category of key frame extraction methods that gains much attention is cluster-based algorithms (Zeng *et al.*, 2008; Shi and Guo, 2010; Cheung and Zakhori, 2005; Peng *et al.*, 2008). Cluster-based algorithms (Peng *et al.*, 2008) require a preprocessing step that transforms frames into points of a feature space, where clustering methods are applied and all points are grouped into a bunch of clusters. A cluster selection step is usually followed by picking up significant clusters and extracting frames that are close to cluster centers as key frames. Cluster-based algorithms rely on a suitable feature space to represent the content of frames. However, good and clean clusters are not easy to be formed and, therefore, the patterns of data points in the feature space are not straightforward. In addition, cluster-based methods are more complicated than the aforementioned key frame extraction methods. Theoretically, inter-cluster visual variance is large; while intra-cluster variance is small. Hence, the redundancy within the extracted key frames can be kept below a certain level.

There are also some other algorithms that focus on addressing the redundancy problem in extracted key frames. One method used the integration of local and global information to remove redundancy in the set of key frame candidates and achieved good results (Liu *et al.*, 2010). Minimum-Correlation based algorithms (Doulamis *et al.*, 2008; Liu and Kender, 2002; Porter *et al.*, 2003) assumed that the key frames had little correlation with each other. By pruning some significantly correlated frames, the algorithms could ensure that the extracted key frames hold a low level of redundancy. However, minimum-correlation based algorithms were vulnerable to outliers.

Furthermore, key frames could be extracted based on clips (Yu *et al.*, 2004; Girgensohn and Boreczky, 1999) or shots (Liu and Fan, 2005; Kelm *et al.*, 2009; Abd-Almageed, 2008; Liu *et al.*, 2009). Shot-based approaches are quite intuitive since shots are regarded as the basic semantic unit in videos. In addition, shot segmentation techniques are quite mature recently, and thus it is applicable to detect shots within a video before applying key frame extraction algorithms. The simplest approach of shot-based key frame extraction is to choose the first frame of each shot as a key frame, which works well for shots with low motion. Divakaran *et al.* (2008) and Narasimha *et al.* (2003) adopted motion changes within a shot as a criterion to select key frames. The idea was that more key frames should be extracted in shots that consist of frequent motion activity changes. However, motion features obtained in Divakaran *et al.* (2008) and Narasimha *et al.* (2003) were from MPGE-7 motion activity descriptors which were not easy to be applied to the uncompressed videos. Therefore, these approaches were limited to the compressed domain. To address such a limitation, Liu *et al.* (2003) built a motion energy model to perceive motion patterns in the uncompressed domain. However, motion features do not always represent major content within shots. Hence, their approaches only work well for a particular field that is highly related with motion, such as sports (Divakaran *et al.*, 2008; Narasimha *et al.*, 2003; Liu *et al.*, 2003).

Compared with motion information, visual content in shot/video is described more reliably by colour features. Mendi and Bayrak (2010) created saliency maps based on colour and luminance features in video frames, and the similarity between frames was calculated by using a new signal fidelity measurement called S-SSIM. Frames with the highest S-SSIM in each shot were extracted as key frames. In Lin *et al.* (2008), coarse regions were first detected, and then interesting points in these detected regions served as a basis to compare similarity between frames at the shot level. Later, one key frame was extracted from each shot. Although Mendi and Bayrak (2010) and Lin *et al.* (2008) reported good experimental results, they still faced the same problem that many frames containing important visual content might not be extracted as key frames, since one key frame per shot is usually insufficient to represent the shot content. In order to completely summarise videos, more key frames need to be extracted from each shot; otherwise, the summarisation quality represented by key frames would be compromised. On the other hand, the increment in the number of frames extracted from each shot will increase redundancy to the final set of key frames. Therefore, the issue to balance the quantity

and the quality in key frame extraction is also a major concern.

3. The Proposed Video Key Frame Extraction Approach

Two main phases are included in the proposed approach. The first one is the cluster-based key frame candidate extraction phase, which extracts a group of key frame candidates (KFCs) on informative regions for each shot from the video sequences. The second one is the filtering phase, in which the information within a video shot and between video shots extracted from KFCs is used to remove those redundant KFCs. Figure 1 presents the system architecture of our proposed key frame extraction approach.

3.1. A difference measure between consecutive frames

One of the frequently used methods that measure the differences of images is based on a colour histogram because of its computation simplicity. A colour histogram is a representation of the distribution of colours in an image. For digital images, a colour histogram is represented by counting the number of pixels belonging to each colour. It provides a compact summarisation of the colour information in an image. On the other hand, the colour histogram loses the spatial distribution information of colour data. For example, by comparing the colour histogram, we can infer that an image is red, but cannot tell which part of the image is red. Considering such a drawback of the colour histogram, in this paper, we use a colour feature vector to represent the video frames and employ the Euclidean distance of the feature vectors to measure the difference between two frames.

For the sake of fast computation, each frame has been partitioned into several squares, and each square is a 16×16 block of pixels (as proposed by Kim *et al.*, 2008). Then the average pixel value of each square is calculated as a feature of that frame. In other words, the frame has been re-sized to $1/256$ of the original size (each dimension being re-sized to $1/16$ of its original size), and each pixel in a new image is represented by the average pixel value of the corresponding 16×16 block in the original frame. For instance, a new image with lower resolution (18×22 pixels) will be generated from an original frame with 288×352 pixels. If the columns are sequentially concatenated, the 18×22 image can be converted into a feature vector with 396 features. Therefore, a frame with 288×352 pixels is projected into a feature space $\mathbb{R}^{396}$ as a feature point with 396 features. Before re-sizing, the original

Fig. 1. System architecture of the proposed key frame extraction approach.

frame is first transferred from RGB colour space to $YCbCr$ colour space, and $YCbCr_{avg}$ was calculated to represent each pixel in the frame by Eq. (1) (Kim *et al.*, 2008):

$$YCbCr_{avg} = \frac{2}{3} \cdot Y + \frac{1}{6} \cdot Cb + \frac{1}{6} \cdot Cr. \quad (1)$$

The problem of estimating the difference in the visual content of two frames now has been converted into a similarity measure of their feature vectors. There are several distance formulas for measuring the similarity of

the feature vector. In this paper, Euclidean distance $d(\cdot)$ is employed to measure the similarity of the two feature vectors $\mathbf{p}$ and $\mathbf{q}$ using Eq. (2):

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}, \quad (2)$$

where $\mathbf{p} = (p_1, p_2, \dots, p_n)$, $\mathbf{q} = (q_1, q_2, \dots, q_n)$, and n is the dimension of the feature vector (e.g., $n = 396$ in the above example).

3.2. A cluster-based frame extraction method on informative regions

A video sequence is made up of a group of shots and each shot is a series of consecutive frames from the start to the end of recording in a camera. For some shots, there is little visual content difference between successive frames. In such cases, a single frame may be sufficient to represent the content of all the frames in the shot. On the other hand, for those shots whose content is more various and complex, more key frames are needed to represent the shots. One of the most common methods to select key frames is the temporal sampling method (Hoon *et al.*, 2000). Though the method is fast and easy to use, it usually does not provide a successful representation since it ignores the fact that the variation of the content within a shot is usually not proportional to the shot length.

In order to effectively select a proper set of key frames to represent the corresponding shot, the frames in each shot are first separated into two types: transitive frames (TFs) or informative frames (IFs). TFs are those frames that have large pictorial content differences compared with their adjacent frames, implying the transition of visual content due to the relatively fast movement of the camera. Compared with TFs, IFs contain more stable content and objects, which are the real visual information that the recorder wants to take. Based on the above assumption, key frames are selected among IFs, but key frames falling within TFs are ignored.

Followed by the difference calculation between adjacent frames, the idea of constructing a binary classifier to identify TFs and IFs from Otsu (1979) is adopted. It comes from the field of computer vision and image processing, and computes a global threshold that can be used to convert a colour image or a grayscale image to a binary image by choosing the threshold to minimise the intra-class variance of the black and white pixels. The main objective of classifying TFs and IFs is to divide a shot into informative regions and transitive regions. A transitive region whose members are mostly TFs contains blurry objects or uniformly coloured images that are meaningless in terms of the information supplied. Therefore, key

frames should be selected from the informative regions. As can be seen from Fig. 2, the first row is an initial classification of IFs and TFs, and the second row shows more continuous regions after a smoothing processing.

For key frame selection, a clustering approach is developed and employed in our proposed approach. Cluster analysis is the formal study of methods and algorithms for grouping data (Jain, 2010). The general idea of cluster-based key frame extraction methods is to consider all the frames in a shot together and cluster them based on the similarity of their feature vectors. The frame that is closest to the cluster's centre is usually selected as a key frame. One of the problems in clustering is that in order to identify a key frame, the centre of a cluster needs to be first calculated. In our proposed approach, this step is omitted in order to reduce the computation time. In other words, instead of calculating the cluster's centre and its distance with nearby frames, our proposed approach utilises the middle frame of each shot as the first KFC f_1 . Based on f_1 , the second KFC f_2 is chosen using the following criterion:

$$\arg \max_{f_2} d(f_1, f_2). \quad (3)$$

The above expression is the set of values of f_2 for which $d(f_1, f_2)$ has the largest value, where $\arg \max$ stands for the argument of the maximum, and $d(f_1, f_2)$ indicates the difference in the visual content between f_1 and f_2 obtained by Eq. (2) in Section 3.1. To generalize the selection rule, the n th KFC f_n is selected by the following criterion:

$$\arg \max_{f_n} \sum_{k=1}^{n-1} d(f_k, f_n). \quad (4)$$

This selection criterion chooses f_n for which the sum of differences between f_n and previous $n - 1$ KFCs ($k = 1, 2, 3, \dots, n - 1$) has the largest value among all non-KFCs in the shot.

3.3. KFC filtering by integrating information within a video shot and between video shots

One of the major issues in key frame extraction is to decide the amount of key frames that should be selected per shot.

Fig. 2. Identification of transitive regions and informative regions.

A large set of key frames would give a more comprehensive representation of the video sequence, but at the same time it would result in content redundancy. On the other hand, a small set of key frames could restrict pictorial redundancy, but it hardly represents the video content completely. Another issue is that the number of extracted key frames in each shot should be varied due to the unequal quantities of information conveyed in different shots. A commonly used method is pre-setting a threshold T in the key frame extractor, but the determination of the value for T is another decisive factor to affect the final performance of the key frame extractor. Chatzigiorgaki and Skodras (2009) used two videos from TRECVID 2007 test dataset (Smeaton et al., 2006) as a training set to conduct the threshold selection process, which achieved good results in their experiments. It is an acceptable approach that employs the training videos to calculate the threshold for key frame extractors, but it would be more inspiring if a self-adapted method can be developed to decide the number of key frames to be extracted in each shot based on the video sequence information itself. Such an extraction method would be more compact and accurate than those that adopt thresholds calculated from other training videos.

In this paper, we utilise the information within a shot and between shots (e.g., standard deviation of KFCs) to filter KFCs. Assume that the shot content changes relatively small, so the value of the standard deviation of KFC's feature vectors in the shot should be small, and vice versa. On the basis of such an assumption, the standard deviation of the whole KFCs set is used as a threshold to measure the content variation in each shot. The strategy is that if the standard deviation of the j th shot is less than the standard deviation of the whole KFCs set of the video, only KFC f_1 is reserved to represent the j th shot. Otherwise, the Euclidean distances between KFCs are used to decide how many KFCs are kept as key frames to represent the j th shot. For instance, using the first two KFCs f_1 and f_2 that have been kept as key frames of the j th shot to decide whether f_3 should be kept as a key frame by evaluating the relationships among $d(f_1, f_2)$, $d(f_1, f_3)$ and $d(f_2, f_3)$. If in the case of Fig. 3(a) that $d(f_1, f_3) + d(f_2, f_3) > 2 * d(f_1, f_2)$, which means f_3 may contain extra information in addition to f_1 and f_2 ; then f_3 will be reserved as a final key frame. However, if f_1 , f_2 and f_3 construct a relation as shown in Fig. 3(b) that $d(f_1, f_3) + d(f_2, f_3) \leq 2 * d(f_1, f_2)$, it implies f_3 may have similarly visual content with f_1 or f_2 ; in this case, f_3 will be removed from set of KFC, reserving f_1 and f_2 as final key frames to represent the j th shot.

Furthermore, in the case of Fig. 4, assuming that f_3 has been selected as a key frame, KFC f_4 will be evaluated to obey the similar strategy. If the average length of the three

Fig. 3. Two kinds of space layouts of f_1 , f_2 and f_3 : (a) $d(f_1, f_3) + d(f_2, f_3) > 2 * d(f_1, f_2)$ and (b) $d(f_1, f_3) + d(f_2, f_3) \leq 2 * d(f_1, f_2)$.

gray lines in Fig. 4 is larger than the average length of the other bold lines, f_4 would be kept as a key frame. The same process applies to evaluate f_5 and so on.

If f_3 is removed, f_4 will not be considered and the process of filtering KFCs in the j th shot terminates. The general rules in the filtering phase are as follows:

```

IF  $std(j) < \alpha \cdot std(video)$  THEN
    Keep KFC  $f_1$  as a key frame, dumping subsequent
    KFCs in the  $j$ th shot
ELSE
    WHILE
        
$$\left( \frac{1}{n-1} \sum_{k=1}^{n-1} d(f_k, f_n) > \frac{(n-1)(n-2)}{2} \sum_{p=1, q=2}^{n-1} d(f_p, f_q) \right)$$

        Keep KFC  $f_n$  as a key frame and  $n = n + 1$ .
    END-WHILE;
END-IF.

```

Here, $std(j)$ denotes the standard deviation of the KFCs in the j th shot, $std(video)$ denotes the standard deviation of all KFCs in the video, and α is the coefficient whose value is between 0 and 1. The expression in the “WHILE” statement indicates that if the average Euclidean distance between KFC f_n and the other $n - 1$ key frames in the same shot is larger than the average Euclidean distance between $n - 1$ key frames, KFC f_n should be kept as a key frame and the process to evaluate KFC f_{n+1} is continued.

4. Experimental Results

Usually, a video sequence is first divided into meaningful segments (shots), and then each shot is represented and summarised by key frames. Shot boundary detection

Fig. 4. The space layout of f_1 , f_2 , f_3 and f_4 .

algorithms aim to break up a video into meaningful sub-segments by using pixel comparison between consecutive frames, edge changes, grey-scale and RGB colour histogram features, and similarity analysis. In this paper, shot boundary information was given by the data provider, so shot boundary detection is not discussed in this paper. Here, we focus on testing the robustness of the proposed key frame extraction method at the shot level.

4.1. Evaluation metrics

We carried out the evaluation of the proposed approach in terms of the percentage of the extracted key frames and the retrieval precision. In the premise of no significant scenario missing during the key frame extraction process, the smaller number of key frames we use to represent the video, the better performance the key frame extractor does. Therefore, low key frame percentage is preferred to avoid unnecessary content redundancy in the set of key frames. The percentage of the extracted key frames (%KF) is defined as follows.

$$\% \text{ KF} = \frac{\text{number of extracted key frames}}{\text{total number of frames}} \times 100\%.$$

In statistics, the precision is the number of correct results divided by the number of all returned results. To define correctly extracted key frames, we introduced a concept called *hit deviation* to evaluate the quality of the extracted key frames. *Hit deviation* is defined as the difference between the true frame (ground truth) index number and the extracted key frame index number. If the distance of the frame index number between a true key frame and an extracted key frame is less than the preset hit deviation threshold, it means a correct extraction. For instance, if the index number of a true key frame is 39, and

Fig. 5. Hit deviation measure of the extracted key frames, where t denotes the frame index number at time t ; KF1, KF2, and KF3 denote the extracted key frames; GT1 and GT2 denote the true key frames (ground truth); and HD1 and HD2 denote the hit deviations.

the hit deviation threshold is set to 5, an extracted key frame's index number between 34 and 44 is viewed as a correct hit. In case two or more extracted key frames hit one true key frame, only the nearest one was recorded and the others are ignored. As shown in Fig. 5, KF2 fails to hit its adjacent ground truths GT1 and GT2, since there are other key frames that are closer to the true key frames GT1 and GT2 than KF2. In the case of MPEG-1 video with 25fps, if the difference between two frames' index numbers is less than 25, it means the time interval between the two frames is less than one second. Generally speaking, the visual content changes would be relatively small within one second and those frames in one second have similar pictorial content.

4.2. Videos data sets

Fourteen MPEG-1 video sequences with 25fps from TRECVID 2007 test video collection (Smeaton *et al.*, 2006) were used to evaluate the performance of our proposed key frame extraction approach. Table 1 summarises

Table 1. The 14 videos used to test the key frame extraction approaches.

Video name	Length (hh:mm:ss)	# of shots	# of Fr	FrPerShot	# of true KF	tKFPerShot
BG_2196	00:26:13	124	39339	317.234	147	1.185
BG_10241	00:15:40	131	23517	179.504	146	1.115
BG_11369	00:06:33	59	9828	166.542	90	1.525
BG_34837	00:15:05	156	22624	145.026	203	1.301
BG_35447	00:14:59	127	22489	177.079	174	1.370
BG_35751	00:15:11	108	22786	210.981	137	1.269
BG_35757	00:14:53	85	22327	262.671	112	1.318
BG_35767	00:14:27	74	71679	968.635	94	1.270
BG_36304	00:18:06	163	27169	166.681	290	1.779
BG_36366	00:14:28	92	21706	235.935	113	1.228
BG_36511	00:10:03	72	15075	209.347	86	1.194
BG_37613	00:15:53	117	23830	203.675	139	1.188
BG_37796	00:15:16	74	22912	309.622	92	1.243
BG_38002	01:08:53	700	103347	147.639	1003	1.433

Table 2. Key frame percentage (%).

%KF	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method	Ground truth
BG_2196	0.946	0.630	0.630	0.493	0.374
BG_10241	1.671	1.114	1.114	0.850	0.621
BG_11369	1.801	1.201	1.201	0.926	0.916
BG_34837	2.069	1.379	1.379	1.224	0.897
BG_35447	1.694	1.129	1.129	1.009	0.774
BG_35751	1.422	0.948	0.948	0.812	0.601
BG_35757	1.142	0.761	0.761	0.703	0.502
BG_35767	0.310	0.206	0.206	0.187	0.131
BG_36304	1.800	1.200	1.200	1.097	1.067
BG_36366	1.272	0.848	0.848	0.691	0.521
BG_36511	1.433	0.955	0.955	0.736	0.570
BG_37613	1.473	0.982	0.982	0.864	0.583
BG_37796	0.969	0.646	0.646	0.554	0.402
BG_38002	2.032	1.355	1.355	1.161	0.971
Average	1.431	0.954	0.954	0.808	0.638

the characteristics of the fourteen video test sequences, including the name, length, number of shots (# of shots), number of frames (# of Fr), average number of frames per shot (FrPerShot), number of ground truths (# of true KF), and average number of ground truths per shot (tKFPPerShot).

In addition to the proposed approach, for the purpose of comparison, we also tested the commonly used key frame extraction method: temporal sampling. In particular, temporal sampling was implemented into two versions. One is average temporal sampling that samples a pre-fixed number of frames per shot with an equal interval. Another sampling method is called adaptive temporal sampling (Hoon *et al.*, 2000) whose initial purpose was to select more frames in a rapidly changing shot region. The method selects sampling rate on the basis of accumulated

value of the colour histogram differences in the video. We use a modified version of adaptive temporal sampling by using our feature vectors in Section 2.1 instead of the colour histogram. *I*-frames of each shot were first selected as basic frames for the KFC extraction.

4.3. Results

The experimental results of the key frame percentages are presented in Table 2, where AvgTS (3) and AvgTS (2) denote the average temporal sampling with sampling rates at 3 and 2 frames per shot, respectively; AdaTS (2) denotes the adaptive temporal sampling with an average sampling rate on two frames per shot. One of the drawbacks in temporal sampling method is that the sampling

Table 3. Precision results when hit deviation (HD) was set to 5.

Precision (HD = 5)	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method
BG_2196	0.129	0.096	0.049	0.191
BG_10241	0.115	0.134	0.088	0.245
BG_11369	0.158	0.144	0.077	0.264
BG_34837	0.167	0.179	0.096	0.213
BG_35447	0.165	0.185	0.083	0.225
BG_35751	0.142	0.162	0.07	0.243
BG_35757	0.129	0.135	0.065	0.204
BG_35767	0.135	0.155	0.034	0.194
BG_36304	0.211	0.242	0.126	0.255
BG_36366	0.145	0.168	0.055	0.253
BG_36511	0.116	0.167	0.084	0.216
BG_37613	0.134	0.115	0.043	0.252
BG_37796	0.153	0.080	0.048	0.212
BG_38002	0.175	0.192	0.096	0.227
Average	0.148	0.154	0.072	0.228

Table 4. Precision results when hit deviation (HD) was set to 15.

Precision (HD = 15)	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method
BG_2196	0.272	0.250	0.113	0.392
BG_10241	0.275	0.302	0.215	0.445
BG_11369	0.384	0.373	0.316	0.440
BG_34837	0.338	0.394	0.273	0.422
BG_35447	0.344	0.406	0.237	0.480
BG_35751	0.278	0.324	0.181	0.476
BG_35757	0.275	0.300	0.160	0.408
BG_35767	0.257	0.318	0.095	0.418
BG_36304	0.470	0.543	0.335	0.567
BG_36366	0.272	0.348	0.158	0.507
BG_36511	0.306	0.34	0.203	0.432
BG_37613	0.308	0.329	0.155	0.456
BG_37796	0.252	0.236	0.122	0.386
BG_38002	0.365	0.433	0.300	0.499
Average	0.314	0.350	0.205	0.452

Table 5. Precision results when hit deviation (HD) was set to 30.

Precision (HD = 30)	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method
BG_2196	0.328	0.403	0.202	0.546
BG_10241	0.331	0.447	0.372	0.565
BG_11369	0.458	0.610	0.573	0.857
BG_34837	0.410	0.593	0.521	0.657
BG_35447	0.430	0.539	0.482	0.665
BG_35751	0.340	0.444	0.340	0.611
BG_35757	0.365	0.447	0.325	0.599
BG_35767	0.311	0.405	0.218	0.560
BG_36304	0.560	0.791	0.683	0.869
BG_36366	0.344	0.473	0.295	0.600
BG_36511	0.347	0.451	0.441	0.604
BG_37613	0.353	0.466	0.361	0.563
BG_37796	0.320	0.392	0.279	0.512
BG_38002	0.447	0.618	0.590	0.733
Average	0.382	0.506	0.406	0.639

Table 6. Precision results when hit deviation (HD) was set to 60.

Precision (HD = 60)	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method
BG_2196	0.366	0.512	0.409	0.660
BG_10241	0.369	0.538	0.556	0.690
BG_11369	0.497	0.712	0.761	0.923
BG_34837	0.429	0.635	0.650	0.722
BG_35447	0.545	0.654	0.684	0.744
BG_35751	0.407	0.569	0.605	0.676
BG_35757	0.412	0.559	0.627	0.643
BG_35767	0.378	0.520	0.531	0.642
BG_36304	0.581	0.856	0.889	0.946
BG_36366	0.370	0.538	0.612	0.660
BG_36511	0.389	0.556	0.762	0.748
BG_37613	0.393	0.557	0.592	0.646
BG_37796	0.365	0.500	0.476	0.591
BG_38002	0.471	0.695	0.717	0.808
Average	0.427	0.600	0.634	0.721

rate should be pre-set manually. Since the average number of true key frames per shot is greater than one, for adaptive temporal sampling, we extracted twice of the number of ground truth key frames to represent the video. For average temporal sampling method, we set the sampling rate to two and three.

As can be seen from the results, the key frame percentages of our proposed approach were all limited to a maximum of 1.224%; while on average it reached 0.808%. In other methods, the maximum is 1.379% or 2.069% and the average key frame percentage is 0.954% or 1.431%. Compared with the ground truths that reaches 1.067% on

Fig. 6. Key frames extracted by average temporal sampling (3 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196.

maximum and 0.638% on average, it shows that our proposed approach effectively eliminates the redundancy and controls the key frame percentage in an acceptable level (i.e. about 1.2 times of the ground truths on average). It indicates that our proposed key frame extraction approach is able to find a smaller subset of frames, which is in fact more representative in summarising the video. This is preferable in any key frame extraction method. Tables 3–6 show the precision of four methods with hit deviation setting at 5, 15, 30 and 60, respectively. When we set the hit deviation to 5, 15, 30 and 60, the best average results of temporal samplings are 15.4%, 35%, 50.6% and 63.4%, in contrast to the average precision values of our proposed approach, namely 22.8%, 45.2%, 63.9% and 72.1%, respectively. It indicates that our proposed approach outperforms the temporal samplings approach no matter the average one or the adaptive one, which verifies the effectiveness of our proposed approach in video summarisation.

In order to make a pictorial comparison on the visual content of the extracted key frames, we selected a video

clip to evaluate the proposed approach and temporal sampling approaches. The extraction results are shown in Figs. 6–10. Compared with the ground truth in Fig. 10, lots of redundant frames existed in Fig. 6. The same problem also happens in Figs. 7 and 8. Furthermore, content redundancy in Figs. 7 and 8 also suffered from the issue of missing key frames. The results of our proposed approach are shown in Fig. 9, which has successfully extracted effective key frames and reduced the overlapped information.

4.4. Time complexity analysis

In terms of the theoretical complexity, the average temporal sampling, adaptive temporal sampling, and our proposed approach take $O(1)$, $O(N)$, and $O(N^2)$, respectively, where N is the number of frames in a video sequence. Though our proposed approach has a higher time complexity than those of the other two approaches, it should not be an issue of concern since in most cases, key frames can be extracted off-line. The fact that our proposed approach achieves lower percentage and better

Fig. 7. Key frames extracted by average temporal sampling (2 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196.

Fig. 8. Key frames extracted by adaptive temporal sampling (averagely 2 frames per shot) on 11 consecutive shots from shot 11 to shot 21 in video BG_2196.

Fig. 9. Key frames extracted by proposed method on 11 consecutive shots from shot 11 to shot 21 in video BG_2196.

Fig. 10. Ground truth on 11 consecutive shots from shot 11 to shot 21 in video BG_2196.

precision of the extracted key frames makes the key frame-based applications, such as search and browsing, become more efficient in terms of time and space complexities. Table 7 shows the computation time (in minutes) of our

Table 7. Computation time of four methods on the 14 videos.

Extraction time (minute)	AvgTS (3)	AvgTS (2)	AdaTS (2)	Proposed method
BG_2196	0.017082	0.016620	1.362779	243.447
BG_10241	0.017331	0.060535	4.431702	166.835
BG_11369	0.007284	0.005452	4.095344	68.483
BG_34837	0.017959	0.010439	1.383119	150.277
BG_35447	0.016773	0.016574	0.563536	142.304
BG_35751	0.017110	0.016050	1.224688	164.526
BG_35757	0.009854	0.015565	0.750621	151.175
BG_35767	0.015513	0.009274	0.733494	142.495
BG_36304	0.020402	0.012377	1.636738	182.165
BG_36366	0.015871	0.015543	1.405718	159.646
BG_36511	0.010955	0.009050	4.134540	99.356
BG_37613	0.017339	0.010614	2.167435	159.450
BG_37796	0.016350	0.015164	0.961948	150.424
BG_38002	0.060810	0.064711	1.760058	733.047
Average	0.018617	0.019855	1.900837	193.831

proposed key frame extraction approach, which was implemented in the Matlab R2008a development environment. The computer used for the comparison was an Intel Core2 Duo CPU T6400 (2.00 GHz) with 4 GB of RAM, and running a Windows 7 Home Premium operating system.

5. Conclusions and Future Work

This paper proposes an effective key frame extraction approach by utilising the information within a video shot and between video shots. Our proposed approach first selects a set of key frame candidates in the informative regions, and then removes a few redundant key frame candidates to finalise the set of key frames based on the evaluation of within and between shot information. Through the filtering process, most of the redundant key frame candidates are successfully deleted to obtain a reduced set of key frames. According to the performance in terms of extraction percentage and retrieval precision, the proposed approach effectively demonstrates its capability of extracting effective key frames while reducing overlapped visual content.

One of our further improvement directions is to self-adaptively choose the position of the initial KFC f_1 , rather than using the middle frame of a shot as the initial KFC f_1 . Another enhancement direction is to extract KFCs by using object and motion information in both temporal and spatial dimensions from the video sequences. We believe that it would deliver compensatory information which is not available in the current image-based key frame extraction methods.

Acknowledgement

For Shu-Ching Chen, this work was supported in part by NSF HRD-0833093.

References

- Abd-Almageed, W (2008). Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing. In *Proceedings of the 15th IEEE International Conference on Image Processing*, San Diego, California, USA, October 12–15, pp. 3200–3203.
- Chang, HS, S Sull and SU Lee (1999). Efficient video indexing scheme for content-based retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), 1269–1279.
- Chatzigiorgaki, M and AN Skodras (2009). Real-time keyframe extraction towards video content identification. In *Proceedings of the 16th International Conference on Digital Signal Processing (DSP2009)*, Santorini, Greece, July 5–7, pp. 934–939.
- Cheung, S-S and A Zakhori (2005). Fast similarity search and clustering of video sequences on the world-wide-web. *IEEE Transaction on Multimedia*, 7(3), 524–537.
- Cooper, M and J Foote (2005). Discriminative techniques for keyframe selection. In *Proceedings of the 2005 International Conference on Multimedia and Expo (ICME 2005)*, Amsterdam, The Netherlands, July 6–9, pp. 502–505.
- Divakaran, A, R Radhakrishnan and KA Peker (2008). Motion activity-based extraction of key-frames from video shots. In *Proceedings of the 15th International Conference on Image Processing*, San Diego, California, USA, October 12–15, Vol. 1, pp. 932–935.
- Doulamis, ND, AD Doulamis, YS Avrithis and SD Kollias (2008). Video content representation using optimal extraction of frames and scenes. In *Proceedings of the 15th IEEE International Conference on Image Processing*, San Diego, California, USA, October 12–15, Vol. 1, pp. 875–879.
- Fu, Y-W, Y-W Guo, YS Zhu, F Liu, CM Song and Z-H Zhou (2010). Multi-view video summarization. *IEEE Transactions on Multimedia*, 12(7), 717–729.
- Girgensohn, A and J Boreczky (1999). Time-constrained keyframe selection technique. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, June 7–11, Vol. 1, pp. 756–761.
- Hoon, SH, K Yoon and I Kweon (2000). A new technique for shot detection and key frames selection in histogram space. In *Proceedings of the 12th Workshop on Image Processing and Image Understanding*, January, pp. 475–479.
- Jain, A (2010). Data clustering: 50 years beyond K -means. *Pattern Recognition Letters*, 31(8), 651–666.
- Kang, H and X Hua (2005). To learn representativeness of video frames. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, Singapore, November 6–12, pp. 423–426.
- Kelm, P, S Schmiedeke and T Sikora (2009). Feature-based video key frame extraction for low quality video sequences. In *Proceedings of the 10th International Workshop on Image Analysis for Multimedia Interactive Services*, London, UK, May 6–8, pp. 25–28.
- Kim, C and J-N Hwang (2002). Object-based video abstraction for video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12), 1128–1138.
- Kim, H, J Lee, H Liu and D Lee (2008). Video linkage: Group based copied video detection. In *Proceedings of the 7th ACM International Conference on Image and Video Retrieval (CIVR'08)*, Niagara Falls, Canada, July 7–9, pp. 397–406.
- Lin, H-H, Y Xuan and J-H Pei (2008). Key frame extraction based on multi-scale phase-based local features. In *Proceedings of the 9th International Conference on Signal Processing (ICSP'08)*, Beijing, China, October 26–29, pp. 1031–1034.
- Liu, L and G Fan (2005). Combined key-frame extraction and object-based video segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7), 869–884.
- Liu, T and JR Kender (2002). Optimization algorithms for the selection of key frame sequences of variable length. In *Proceedings of the 7th European Conference on Computer Vision*, Copenhagen, Denmark, May 27–June 2, pp. 403–417.
- Liu, T, H Zhang and F Qi (2003). A novel video key-frame-extraction algorithm based on perceived motion energy model. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(10), 1006–1013.
- Liu, G-T, X-M Wen, W Zheng and P-Z He (2009). Shot boundary detection and keyframe extraction based on scale invariant feature transform. In *Proceedings of 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2009)*, Shanghai, China, June 1–3, pp. 1126–1130.
- Liu, D, M-L Shyu, C Chen and S-C Chen (2010). Integration of global and local information in videos for key

- frame extraction. In *Proceedings of the 11th IEEE International Conference on Information Reuse and Integration (IRI 2010)*, Tuscany Suites & Casino, Las Vegas, Nevada, USA, August 4–6, pp. 171–176.
- Mendi, E and C Bayrak (2010). Shot boundary detection and key frame extraction using salient region detection and structural similarity. In *Proceedings of the 48th ACM Southeast Conference (ACM-SE'10)*, Oxford, Mississippi, USA, April 15–17.
- Narasimha, R, A Savakis, RM Rao and R De Queiroz (2003). Key frame extraction using MPEG-7 motion descriptors. In *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, USA, November 9–12, Vol. 2, pp. 1575–1579.
- Otsu, N (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1), 62–66.
- Peng, Y, J Pei and Y Xuan (2008). Behavior key frame extraction using invariant moment and unsupervised clustering. In *Proceedings of the 7th International Conference on Machine Learning and Cybernetics*, Kunming, China, July 12–15, pp. 2503–2508.
- Porter, SV, M Mirmehdi and BT Thomas (2003). A shortest path representation for video summarization. In *Proceedings of the 12th International Conference on Image Analysis and Processing (ICIAP'03)*, Mantova, Italy, September 17–19, pp. 460–465.
- Rasheed, Z and M Shah (2003). Scene detection in Hollywood movies and TV shows. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Monona Terrace Convention Center, Madison, Wisconsin, USA, June 16–22, Vol. 2, pp. 343–350.
- Rong, J, W Jin and L Wu (2004). Automatic video summarization. In *Proceedings of the 2004 International Conference on Multimedia and Expo (ICME 2004)*, Taipei, Taiwan, June 27–30, pp. 571–574.
- Shi, F and X Guo (2010). Keyframe extraction based on K-means results to adjacent DC images similarity. In *Proceedings of the 2nd International Conference on Signal Processing Systems*, Dalian, China, July 5–7, pp. 611–613.
- Smeaton, AF, P Over and W Kraaij (2006). Evaluation campaigns and TRECVID. In *8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*, Santa Barbara, California, USA, October 23–27, pp. 321–330.
- Truong, BT and S Venkatesh (2007). Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1), 1–37.
- Yahiaoui, I, B Merialdo and B Huet (2001). Automatic video summarization. In *Multimedia Content-Based Indexing and Retrieval (MMCBIR'01)*, Rocquencourt, France, September.
- Yu, X-D, L Wang, Q Tian and P Xue (2004). Multi-level video representation with application to keyframe extraction. In *Proceedings of International Conference on Multimedia Modeling*, Brisbane, Australia, January 5–7, pp. 117–121.
- Zeng, X-L, W-M Hu, W-Q Li, X-Q Zhang and B Xu (2008). Key-frame extraction using dominant-set clustering. In *Proceedings of the 2008 International Conference on Multimedia and Expo (ICME 2008)*, Hannover, Germany, June 23–26, pp. 1285–1288.
- Zhang, H, J Wu, D Zhong and S Smoliar (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4), 643–658.
- Zhang, X-D, T-Y Liu, K-T Lo and J Feng (2003). Dynamic selection and effective compression of key frames for video abstraction. *Pattern Recognition*, 24(9–10), 1523–1532.
-