New Framework for Automated Extraction of Key Frames from Compressed Video

Kang-Wook Kim[†], Seong-Geun Kwon[††]

ABSTRACT

The effective extraction of key frames from a video stream is an essential task for summarizing and representing the content of a video. Accordingly, this paper proposes a new and fast method for extracting key frames from a compressed video. In the proposed approach, after the entire video sequence has been segmented into elementary content units, called shots, key frame extraction is performed by first assigning the number of key frames to each shot, and then distributing the key frames over the shot using a probabilistic approach to locate the optimal position of the key frames. The main advantage of the proposed method is that no time-consuming computations are needed for distributing the key frames within the shots and the procedure for key frame extraction is completely automatic. Furthermore, the set of key frames is independent of any subjective thresholds or manually set parameters.

Key words: Key Frame, Extraction, Compressed Video

1. INTRODUCTION

Many services, such as VOD (video on demand) and pay television, are provided in digital form to consumers and a rapidly increasing number of interactive multimedia documents, including text, audio, and video, are now available. Consequently, it is widely recognized that there is a need for intelligent management and search methods particularly for visual information in multimedia documents and digital videos. Efficient access to video data located in a distributed database is a very difficult task, mainly due to the large bandwidth requirements imposed by the large amount of video information. Traditionally, video is represented by numerous consecutive frames, each of which corresponds to a constant time interval. However, such a representation is not adequate for new emerging multimedia applications, such as content-based indexing, retrieval, and browsing. Furthermore, tools and algorithms for the effective organization and management of video archives are still limited [1]. There is also an essential need to automatically extract key information from images and videos for the purpose of indexing, fast and easy retrieval, and scene analysis. In order to allow the user to efficiently browse, select, and retrieve a desired video part without having to deal directly with GBytes of compressed data, several activities have to be carried out in preparation for such a user interaction. For videos, a common first step is to segment the videos into temporal "shots," each representing an event or continuous sequence of actions. A shot is what is captured by the camera between a record and a stop operation. Further scene analysis and interpretation can then be performed on such shots. Segmented video sequences can also be used for browsing, in which only one or a few representative frames, i.e., key frames of each shot are displayed [2-5]. The main goal of the

※ Corresponding Author : Seong-Geun Kwon, Address : (712-701) 33 Buho-ri, Hayang-eup, Gyeongsan-si, Gyeongbuk, Korea, TEL : +82-53-850-7158, FAX : +82-53-850-7603, E-mail : sgkwon@kiu.ac.kr

Receipt date : Dec. 12, 2011, Revision date : Mar. 8, 2012
Approval date : Apr. 9, 2012

[†] Samsung Electronics
(E-mail: ekans999@gmail.com)
[††] Department of Electronic Engineering, Kyungil Univ.

above procedures is to provide the user a compact and easily understandable overview of the complete stored video information. Most existing approaches to key frame extraction [6-8], based on measuring the differences between the last selected frame and the remaining frames and extracting a subsequent key frame if the measured difference exceeds the given threshold, are typically sequential processes leading to unpredictable results. Particularly, the final number of key frames for entire sequence can't be estimated and a large number of key frames or too few key frames can be allocated. This makes it difficult to predict the capacity needed to store extracted key frames. Moreover, the dependency on subjective and usually data dependent thresholds, limits its applicability in fully automated systems and leads to bad results.

This paper proposes a new and fast method for extracting key frames from a compressed video. The proposed algorithm can operate directly on Motion JPEG or MPEG compressed video. After the entire video sequence is segmented into elementary content units, called shots, key frame extraction is performed by first assigning the number of key frames to each shot and then distributing the key frames using a probabilistic approach to locate the optimal position of the key frames. The main advantage of the proposed method is that no time-exhaustive computations are needed for distributing the key frames over the shot, plus the procedure of key frame extraction is fully automatic. In addition, the set of key frames is independent of any subjective thresholds or manually given parameters.

In section 2, we explain the concept of our proposed framework for key frame extraction step by step along with simple test results. Experimental results on various video sequences are presented in section 3, demonstrating the performance and validity of the proposed method. Lastly, section 4 gives some final conclusions.

2. PROPOSED KEY FRAME EXTRACTION ALGORITHM

We propose a new three-step key frame extraction method for efficient video content representation. A block diagram of the proposed architecture is illustrated in Fig. 1, and consists of three modules: video segmentation, key frame allocation, and key frame distribution. These three modules are described in the current section. First the video sequence is segmented into distinct video shots, then a mathematical analysis of the video information flow is applied to the frames of each shot. Such an approach provides a more meaningful description of the video content, therefore, the key frame extraction can be implemented more efficiently.

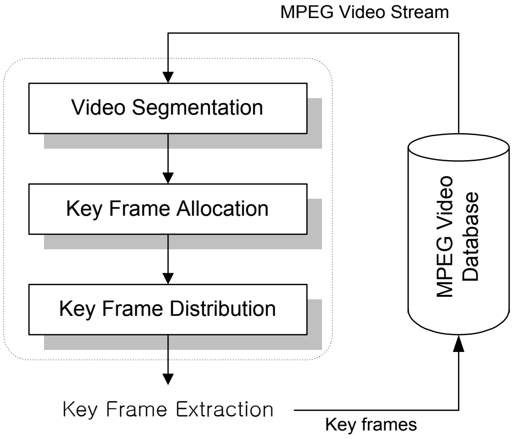


Fig. 1. Block diagram of the proposed architecture.

2.1 Video Segmentation Using DC Image

A DC image is obtained from block-wise averages of 8×8 block. For the I frame of an MPEG coded video, each pixel in the DC image corresponds to a scaled value of the DC coefficient of each DCT block. Thus, each DC image is reduced 64 times compared to the original image. Several algorithms to extract DC images from MPEG compressed video by using DCT DC coefficients in I type frame and motion compensated DCT DC coefficients in P or B type frame were already proposed [9-11].

After extracting DC images from MPEG compressed video, we should detect cut, i.e., shot boundary to segment video into shot. For minimizing the influence of non-relevant temporal variations, global frame visual features such as color and intensity histograms should be used to detect shot boundary. In our approach we adapted the method proposed in [9] and defined an activity function AF(k) for describing the relevant difference between frames k and $k-1$ as:

$$AF(k)=\sum_{i}\sum_{j}\left|I_{DC}^{k}(i,j)-I_{DC}^{k-1}(i,j)\right| \tag{1}$$

where k is the frame index, and $I_{DC}^{k}(i,j)$ means the pixel value at (i,j) position of DC image. $AF(k)$ can measure relative changes between each two consecutive frames and its value indicates the magnitude of such changes. We use $AF(k)$ curve to detect cut as illustrated in [9]. The method of [9] uses a sliding window to examine a few successive frame differences. We declare a scene change from frame $K-1$ to frame k if

1) $AF(k)$ is the maximum within a sliding window of size $2W$, and

2) $AF(k)$ is n times of the second largest maximum in the sliding window.

W is set to be smaller than the minimum duration between two scene changes. For example, setting $W=15$ for a 15 frames/s video means that there cannot be two scene changes within a second. It has been found that values of n ranging form 2.0-3.0 give good result. This method would reduce false detection in cases of significant object or camera motions. Fig. 2 illustrates the plot of $AF(k)$ versus k or the clip of KBS (Korean Broadcasting System) TV news program, specific 1000 frames. From $AF(k)$ curve, we can easily know that this video sequence consists of 7 shots.

If the entire video sequence is segmented into shots by the above mentioned method, the next step is that we should properly assign the number of key frames to each shot and then distribute the key frames over the shot. In the following sections, we will refer to these procedures.

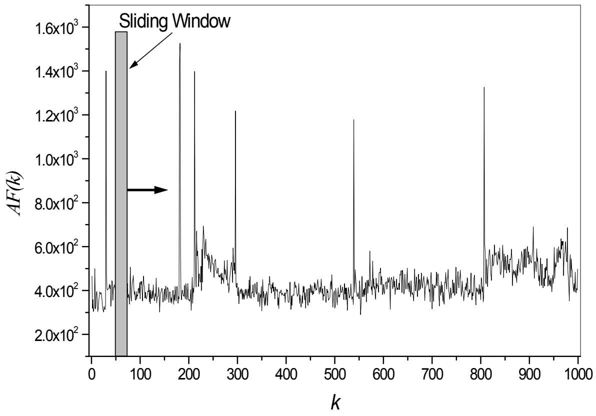


Fig. 2. Plot of $AF(k)$ versus k for 1000 frames.

2.2 Key Frame Allocation to a Shot

We propose a simple intuitively appealing algorithm for finding the number of key frames allocated in each shot. This algorithm allocates key frames to shot incrementally, one key frame at a time, in a way that yields good assignments.

The basic idea is that in each of a total of K_T key frames, one key frame is allocated where it will do the most good at this point. Let $M_i(K_i)$, called the content function, denote the content of the i-th shot for the key frame allocation of K_i key frames. The content function of each shot is defined by

$$M_i(K_i)=CAF_i(L)2^{-2(K_i-1)} \tag{2}$$

, where $CAF_i(n)$ is the accumulated value of $AF(k)$ from the beginning up to the final summation position n. $CAF_i(n)$ can be calculated as follows:

$$CAF_i(n)=\sum_{k=1}^{n}AF(k) \tag{3}$$

, where i, k are the shot and frame index, respectively. If the summation of eq. (3) stretches through the entire frame within a shot, the total magnitude of temporal flow fluctuation in the shot is obtained which represents the content of the shot. In $CAF_i(L)$ of eq. (2), L is the number of frames in the shot.

Let $K_i(m)$ denote the total number of key frames allocated to the i-th shot after iteration m, i.e., after m key frames have been allocated to the shots. Now the request $Q_i(m)$ associated with the i-th shot after the m-th iteration of the allocation algorithm can be defined according to:

$$Q_i(m) = M_i(K_i(m)) \tag{4}$$

That is, the request $Q_i(m)$ after the m-th key frame has been assigned is simply the content of the i-th shot as regards its current key frames. The proposed algorithm assigns K_i key frames to shot i as below.

Step 0. Initialize the key frame allocation to one, so that $K_i(0)=1$ for each i-th shot and $m=0$. Set $Q_i(0)=M_i(K_i(0))$ as the initial values for request.
(The reason for $K_i(0)=1$ is that at least one key frame must be allocated to each shot)

Step 1. Find the shot index j with the maximum request.

Step 2. Set $K_j(m+1)=K_j(m)+1$ and set $K_i(m+1)=K_i(m)$ for each $i \neq j$, then set $Q_i(m+1)=M_i(K_i(m+1))$

Step 3. If $m < K_T - T - 1$, increment m by 1 and go to step 1. Otherwise stop.

T is the number of shots in the entire sequence. This algorithm carries out a very simple and intuitive idea. That is, simply give away key frames to the most needy shot, one key frame at a time until you run out of key frames to give. The degree of neediness of each shot is measured based on the content it will yield if it were to operate with its current key frame assignment. By spreading the given maximal number of key frames K_T along the entire video sequence, each shot of the sequence gets assigned a fraction of the given K_T key frames according to its share of the content relative to the total content of the sequence. Table 1 illustrates the result of key frame allocation for $AF(k)$ curve in Fig. 2.

2.3 Video Segmentation Using DC Image

Here, $l_u(u=1,\cdots,K_i)$ are the temporal locations of the key frames, while n_{u-1} and n_u are the break points between the shot segments represented by key frame l_u. Notice that n_0 and n_{K_i} are the known temporal beginning and end points of the ith shot. The basic idea can be seen in Fig. 3 with K_i assigned key frames.

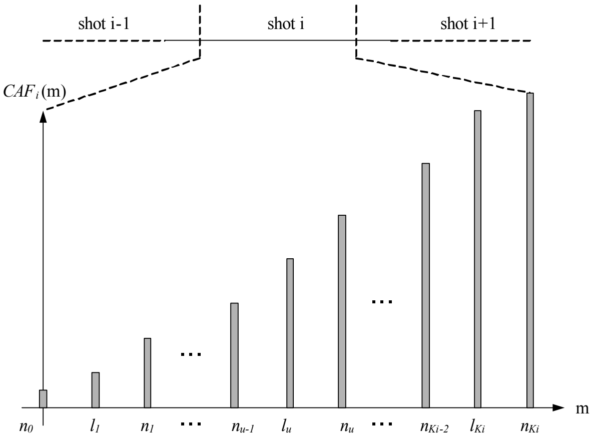


Fig. 3. Key frame distribution within I th shot by assigned K_i key frames.

To find he position of $l_u(u=1,\cdots,K_i)$, we propose a fast and effective method which uses a probabilistic approach to locate the optimal position of the key frames. First, the normalized

Table 1. Results of key frame allocation. ($K_T=10$, $T=7$)

Shot index i	1	2	3	4	5	6	7
K_i	1	1	1	1	2	2	2
L	30	152	30	84	243	268	193
$M_i(K_i(0))$	10,214	60,090	11,304	42,102	93,764	11,168	98,530

$CAF_i(m)(=NCAF_i(m))$ is calculated for the ith shot, which is assumed to be composed of $n_{K_i} - n_0 + 1$ frames between frame n_0 and n_{K_i}. $NCAF_i(m)$ is computed as follows:

$$NCAF_i(m) = Int\left[n_0 + (n_{K_i} - n_0)\frac{CAF_i(m) - CAF_i(n_0)}{CAF_i(n_{K_i}) - CAF_i(n_0)}\right] \\ , \quad m = n_0, \cdots, n_{K_i} \qquad (5)$$

, where $Int[x]$ represents the integer part of x. Using Eq. (5), the discrete $CAF_i(m)$ values that are not interpolated are normalized into integer values lying between the interval $[n_0, n_{K_i}]$. Next, the histogram $H(m)$ of $NCAF_i(m)$ is calculated, then the pmf (probability mass function) $P(m)$ and cdf(cumulative density function) $F(m)$ can be obtained from $H(m)$ using the following relations:

$$P(m) = \frac{H(m)}{n_{K_i} - n_0 + 1},$$

$$F(m) = \sum_{\alpha = n_0}^{m} P(\alpha), \quad m = n_0, \cdots, n_{K_i} \qquad (6)$$

The pmf $P(m)$ is referred to as the probability of change in the shot content. Consequently, only $H(m)$, $P(m)$, and $F(m)$ need to be calculated before distributing the key frames. The remaining key frame distribution procedure is performed by first computing the value q_u such that $F(q_u) = u/K_i$ then finding $n_u = x$ such that $NCAF(x) = q_u$ for $u = 1, \cdots, K_i$. From the above computed n_u, the key frame positions can be easily decided sequentially as follows:

$$l_u = \frac{n_u + n_{u-1}}{2}, \quad u = 1, \cdots, K_i \qquad (7)$$

, where n_0 and n_{K_i} are the known temporal beginning and end points of each ith shot.

This procedure of distributing K_i key frames over the ith shot is very simple and fast. In addition, the proposed method does not require any recursive computations and is performed sequentially. It is intended that the given key frames are distributed over the shot according to the probability of a change in the shot content. Fig. 4 illustrates a summary of the steps involved in the proposed algorithm.

Fig. 5 illustrates the plot of $CAF_i(m)$ versus m for the same sequence as used in Fig. 2. This plot shows the level of content variation for each shot. Therefore, based on the slope and relative magnitude, the importance of each shot can be estimated. The steeper parts correspond to more substantial changes, whereas and flatter parts indicate a more stationary shot variation. The proposed algorithm

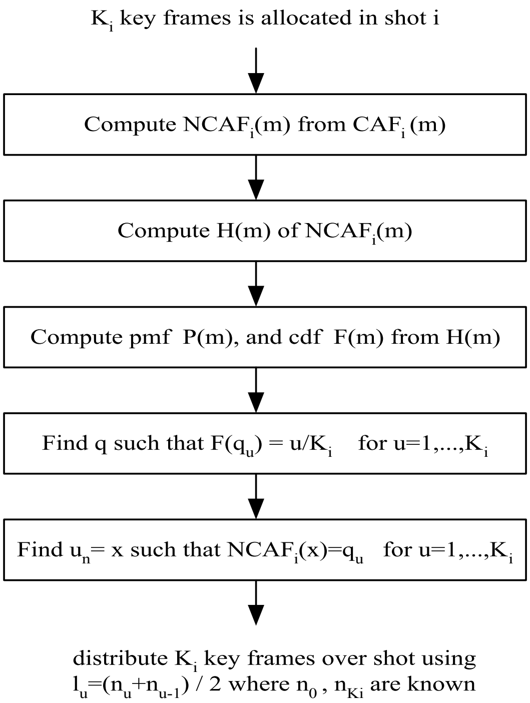

K_i key frames is allocated in shot i
Compute $NCAF_i(m)$ from CAF_i (m)
Compute H(m) of $NCAF_i(m)$
Compute pmf P(m), and cdf F(m) from H(m)
Find q such that $F(q_u)$ = u/K_i for u=1,...,K_i
Find u_n= x such that $NCAF_i(x)=q_u$ for u=1,...,K_i
distribute K_i key frames over shot using
$l_u=(n_u+n_{u-1})$ / 2 where n_0 , n_{Ki} are known

Fig. 4. Flow chart of proposed key frame distribution algorithm.

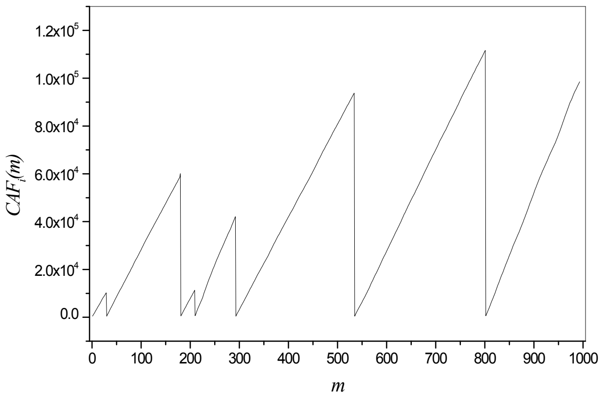


Fig. 5. Plot of $CAF_i(m)$ versus m for test sequence.

Table 2. Results of key frame distribution for 7 shots

Shot index i	1	2	3	4	5	6	7
K_i	1	1	1	1	2	2	2
$l_u(u=1,\cdots,K_i)$	15	106	197	254	357, 478	606, 739	855, 951

is then used to locate the key frames. The result of the key frame distribution is shown in Table 2. The key frames are arranged in a temporal order and extracted in a content-based manner, instead of just simple sub-sampling. For a shot with little or no variation, one key frame (e.g. the first frame) is sufficient. Yet for a long shot or shot with a lot of variations, multiple key frames are chosen. Table 2 shows the effective condensing of 1000 frames of a TV news video clip into 10 key frames. For shots 1-4, only one key frame represents the content of each shot, however, for shots 5-7, two key frames are selected.

Table 3. Video sequences used in experiments

Video sequences	No. of frames	Bit rate	min:sec
TV news ("news.mpg")	10,000	1.300 Mbps	5:33
music video ("music.mpg")	12,541	1.394 Mbps	6:58
sports ("sport.mpg")	20,000	1.300 Mbps	11:07

3. SIMULATIONS

The proposed key frame extraction method was validated by experiment using several long video sequences, as listed in Table 3. The test data were digitized at a 352×240(SIF) spatial resolution from consumer-grade video recordings of TV broadcasts and then compressed in MPEG-1 format at 30 frame/s. The sequences were also available as DC sequences, obtained from MPEG streams with (slightly modified) frame sizes of 44×30.

The reduced DC sequences were first extracted using the algorithm described in section 2.1. Next, the shot boundaries were detected using the method from section 2.1. Generally, recall and precision are used as performance criteria for shot boundary detection methods [11]. The results of the video segmentation are presented in Table 4.

Key frame extraction was then performed using the individual shots obtained after video segmentation. In the experiments, the only parameter set was the maximal number of key frames. Table 5

Table 4. Results of video segmentation

Video sequences	No. of frames	No. of shots T	Recall / Precision (%)	$K_T(=T\times 1.5)$
TV news	10,000	24	91.53 / 84.43	36
Music video	12,391	68	89.92 / 83.67	102
Sports	20,000	53	90.29 / 86.62	80

Table 5. Comparison of performance

% selected frames	Proposed method		Bede Liu's method	
	time	*dissimilarity,* P_T	*time*	*dissimilarity,* P_T
1 % (100 K-frames)	25 s	846.88	110 s	679.95
2 % (200 K-frames)	56 s	1278.45	202 s	1203.27
3 % (300 K-frames)	78 s	1187.38	310 s	1033.96
4 % (400 K-frames)	110 s	1110.03	443 s	927.11
5 % (500 K-frames)	135 s	1153.69	556 s	876.14

depicts the key frame extraction results obtained for the test sequences. The maximal number of key frames K_T was set at 1.5 times the number of shots T for each sequence. However, K_T can be adjusted by the user according to a pictorial summary. Unlike scene change detection, it is hard to define an objective performance analysis method for the assessment of a key frame extraction algorithm. To objectively assess the performance of our proposed method and to compare it with existing method, we define a criterion function P_{S_i} as Eq. (8) in our own way and refer to it as key frame dissimilarity.

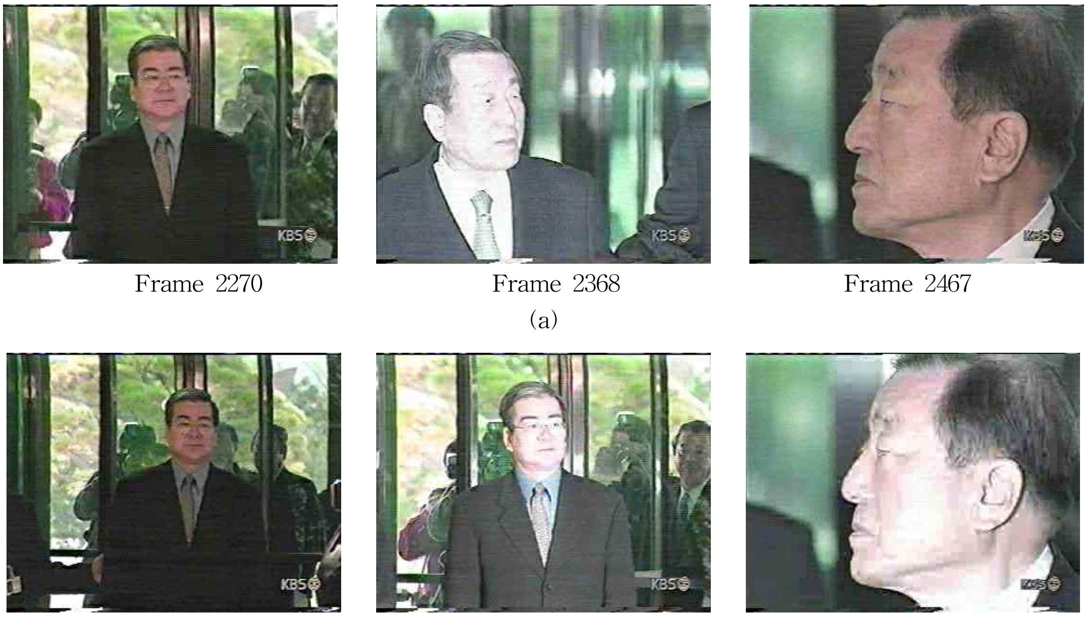
$$P_{S_i} = \frac{1}{K_i - 1}\sum_{j=1}^{K_i-1}\left\{\sum_{m=1}^{M}\sum_{n=1}^{N}\left|f_{key}^{l_{j+1}}(m,n) - f_{key}^{l_j}(m,n)\right|\right\} \quad (8)$$

where S_i means the ith shot and $f_{key}^{l_j}(m,n)$ is the pixel value at position (m,n) of $M \times N$ key frame at temporal location l_j. If $K_i = 1$ in Eq. (8), we take P_{S_i} as 0. Then, we define $P_T = \Sigma P_{S_i}$ as the overall performance measure of key frame extraction method for entire sequence. The smaller value P_T has, the more similar selected key frames are. Performance test has been performed in the following way. The proposed method is compared with Bede Liu's method [9,11], which is typical scheme for extracting key frame. P_T is a better indicator of performance as similarity or dissimilarity between selected key frames. The results are shown in Table 5.

When 5% of sequence is selected as key frames, the proposed method shows 1153.69 of P_T, while the Bede Liu's method shows 876.14 of P_T. It is shown that the key frames selected by proposed method are more dissimilar on another than those chosen by Bede Liu's method. Shown in Fig. 6 are extracted key frames by using proposed and Bede Liu's method. As shown in Fig. 6, proposed method is superior to Bede Liu's method in respect to dissimilarity measure and a subjective point of view for shot 8 for 5% selected frames.

4. CONCLUSIONS

This thesis proposes a new key frame extraction method for content-based video indexing and



Frame 2270 Frame 2368 Frame 2467

(a)

Frame 2221 Frame 2296 Frame 2469

(b)

Fig. 6. Extracted key frames for shot 8 (a) Proposed method ($P_{S_8} = 65.53$) (b) Bede Liu's method ($P_{S_8} = 61.32$).

retrieval. The proposed method consists of three steps: video segmentation, key frame allocation, key frame distribution. The main advantage of the proposed method is that no time-exhaustive computations are needed for distributing the key frames over the shot, plus the procedure of key frame extraction is fully automatic.

REFERENCES

[1] Zeeshan Rasheed and Mubarak Shah, "Detection and Representation of Scenes in Videos," *IEEE Trans. on Multimedia*, Vol.7, No.6, pp. 1097-1105, 2005.

[2] Lijie Liu, "Combined Key-Frame Extraction and Object-Based Video Segmentation," *IEEE Trans. on CSVT,* Vol.15, No.7, 2005.

[3] Jian-quan Ouyang, "Interactive Key Frame Selection Model," *Journal of Visual Commun. and Image Representation*, Vol.17, Issue 6, pp. 1145-1163, 2006.

[4] Lang Congyan, "Automatic Key-Frames Extraction to Represent a Video," *Proc. of IEEE ICSP'04*, Vol.1, pp. 741-744, 2004.

[5] Guozhu Liu and Junming Zhao, "Key Frame Extraction from MPEG Video Stream," *Proc. of IEEE ISIP*, pp. 423-427, 2010.

[6] Kin-Wai Sze, "A New Key Frame Representation for Video Segment Retrieval," *IEEE Trans. on CSVT,* Vol.15, Issue 9, pp. 1148-1155, 2005.

[7] Sang Hyun Kim and Rae-Hong Park, "A Novel Approach to Video Sequence Matching Using Color and Edge Features with The Modified Hausdorff Distance," *Proc. of ISCAS*, pp. II-57~II-60, 2004.

[8] Janko Calic and Ebroul Izquierdo, "Efficient Key-Frame Extraction and Video Analysis," *Proc. of ITCC*, pp. 28-33, 2002.

[9] B.L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video," *IEEE Trans. on CSVT,* Vol.5, No.6, pp. 533-544, 1995.

[10] Fangxia Shi and Xiaojun Guo, "Keyframe Extraction Based on K-means Results to Adjacent DC Images Similarity," *Proc. of ICSPS*, pp. V1-611~V1-613, 2010.

[11] B.L. Yeo and B. Liu, "Fast Extraction of Spatially Reduced Image Sequence from MPEG-2 Compressed Video," *IEEE Trans. on CSVT,* Vol.9, No.7, pp. 1100-1114 1999.

[12] K.W. Kim, J.S. Lee, and S.G. Kwon, "Key Frame Assignment for Compressed Video Based on DC Image Activity," *Journal of Korea Multimedia Society*, Vol.14, No.9, pp. 1109-1116, 2011.

Kang-Wook Kim

He received the B.S., M.S. and Ph. D. degrees in Electronics Engineering, Kyungpook National University, Korea in 1996, 1998 and 2002 respectively. He is currently a senior engineer in R&D Group, Mobile Communication Division, Samsung Electronics Co.,Ltd. His research interests include visual communication, image processing and mobile communication.

Seong-Geun Kwon

He received the B.S., M.S. and Ph. D. degrees in Electronics Engineering, Kyungpook National University, Korea in 1996, 1998, and 2002 respectively. He is currently an assistant professor of the Dept. of Electronic engineering in Kyungil University, Korea. His research interests include mobile broadcasting, watermarking and multimedia security.