Journal of Korea Multimedia Society Vol. 14, No. 9, September 2011(pp. 1109-1116)

Key Frame Assignment for Compressed Video Based on DC Image Activity

Kang-Wook Kim†, Jae-Seung Lee††, Seong-Geun Kwon†††

ABSTRACT

In this paper, we propose a new and fast method for assigning the number of key frames to each shot. At first we segment the entire video sequence into elementary content unit called shots and then the key frame allocation is performed by calculating the accumulated value of AF(activity function). The proposed algorithm is based on the amount of content variation using DC images extracted from compressed video. By assigning the number of key frames to the shot that has the largest value of content function, one key frame is assigned at a time until you run out of given all key frames. The main advantage of our proposed method is that we do not need to use time-exhaustive computations in allocating the key frames over the shot and can perform it fully automatically.

Key words: Key Frame, DC Image, Compressed Video

1. INTRODUCTION

With the integration of information from various and distributed sources and emergence of digital library, browsing of multimedia information in the form of still images and videos will be an important feature of any interactive multimedia systems. During the same time, many services such as VOD (video on demand) and pay television are provided in digital form to the 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 video.

※ 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 : Apr. 29, 2011, Revision date : July 5, 2011
Approval date : July 28, 2011
† Samsung Electronics
(E-mail: ekans999@gmail.com)
†† Korea Aerospace Research Institute
(E-mail: jaeseung.lee@gmail.com)
††† Department of Electronic Engineering, Kyungil Univ.

However, content-based indexing tools and algorithms for the effective organization and management of video archives are still limited. 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, a common first step is to segment the videos into temporal shots, each representing an event or continuous sequence of actions. Next, segmented shots are used for browsing and indexing, in which only one or a few representative frames, i.e., key frames of each shot are displayed [1-5]. To extract key frames from a shot, it is important to properly decide on the number of key frames. This is not an easy or automatic task because the decision is subjective to each person. Accordingly, the question of how many key frames we extract becomes a research topic of importance. 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. In particular, since the final number of key frames for

an entire sequence cannot be estimated, either too large a number of key frames or too few key frames can be allocated, which is ineffective for indexing and browsing. This also makes it difficult to predict the capacity needed to store the extracted key frames in spite of reducing and organizing already obtained key frames. In order to solve these problems, this paper proposes an objective and intuitively appealing algorithm for deciding on the number of key frames allocated to each shot.

In this paper, we propose a new and fast method for assigning the number of key frames to each shot of compressed video. Our algorithm operates directly on Motion JPEG or MPEG compressed video. After we segment the entire video sequence into elementary content unit called shots, the key frame allocation is performed by measuring he accumulated value of activity function. 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. The main advantage of our proposed method is that time-exhaustive computations are not needed in allocating the key frames over the shot and it is performed fully automatically. In addition, this method is not dependent on subjective thresholds or any manually given parameters.

In section 2, we present the concept of video segmentation using DC images and our proposed approach to key frame allocation method. Then, experimental results on various video sequences are presented in section 3, demonstrating the performance and validity of the proposed method. Finally, section 4 concludes the paper.

2. KEY FRAME ASSIGNMENT USING DC IMAGE

2.1 Video Segmentation Using DC Image

DC images are spatially reduced versions of the original images. Such spatially reduced images, once extracted, can also be used for other applications beyond scene change detection. They are used for efficient comparison of video shots, for automatic generation of a compact documents, and for nonlinear video browsing applications. In this section, we briefly show how DC image and DC sequence can be efficiently extracted from compressed videos, and illustrate why they are useful for fast and efficient video segmentation operations. MPEG video stream is generally composed of I, P, and B type frames. 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. It is also hard to extract the DC images from P and B frames, which are coded using motion compensation to use temporal redundancy of video. A general situation is shown in Fig. 1. Here, P_{ref} is the current block of interest, $P_0,\cdots,P_3$ are the four original neighboring blocks from which P_{ref} is derived and the motion vector is $(\triangle x,\triangle y)$.

The shaded regions in $P_0,\cdots,P_3$ are moved by $(\triangle x,\triangle y)$. Our objective is to derive the DC coefficients of P_{ref} . Defining the 2D DCT of an 8 ×8

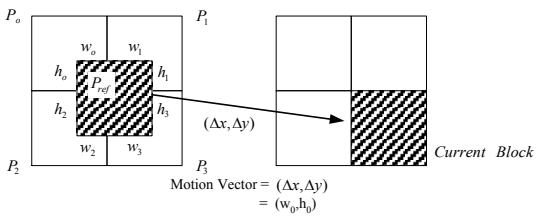

Fig. 1. Reference block (P_{ref}), motion vectors and original blocks.

block P as $DCT(P)$, the linearity of DCT operations mean we can express the DC coefficient of $DCT(P_{ref})$ as:

$$(DCT(P_{ref}))_{00} = \sum_{i=0}^{3}\left(\sum_{m=0}^{7}\sum_{l=0}^{7} w_{ml}^{i}(DCT(P_i))_{ml}\right) \quad (1)$$

, for some weighting coefficients w_{ml}^{i}. The weights w_{00}^{i} is the ratio of overlaps of the block P_{ref} with block P_i, i.e., $w_{00}^{i} = h_i w_i/64$. In this equation, h_i and w_i are the height and width of the overlap of P_{ref} with P_i. An approximation, called the first-order approximation, approximates $(DCT(P_{ref}))_{00}$ by

$$dc(P_{ref})^1 = \sum_{i=0}^{3}\frac{h_i w_i}{64} dc(P_i) \quad (2)$$

When this approximation is applied to B and P frames, it gives good results in practice. More details can be found in [11]. Such approximation requires only the motion vector information and DC values in the reference frames. 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]. We illustrate in Fig. 2 an original image of size 352 ×240 and its DC image of size 44 ×30.

It is demonstrated that even at this low resolution, global image features useful for specific class of content-based operations on MPEG compressed video streams are well preserved. 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:

$$AF(k) = \sum_i \sum_j \left| I_{DC}^{k}(i,j) - I_{DC}^{k-1}(i,j) \right| \quad (3)$$

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. 3 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 properly should assign the number of key frames to each shot and then distribute the

Fig. 2. Full image at 352×240 and its DC image at 44×30.

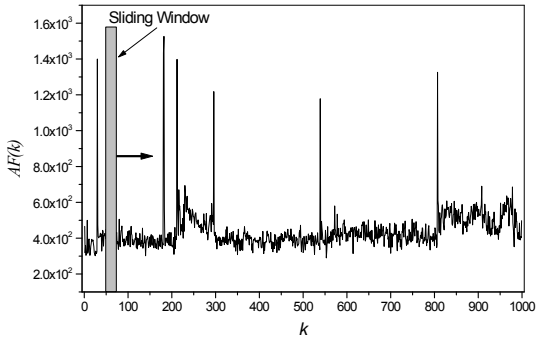

Fig. 3. Plot of $AF(k)$ versus k for 1000 frames.

key frames over the shot. In the following sections, we will refer to these procedures.

2.2 Key Frame Allocation to a Shot

To represent video shots, it is necessary to properly decide the number of key frames (or representative frames) and select these key frames form each shot. Generally, it is not an easy task to determine the number of key frames automatically because the determination is very subjective to each person. Selecting one key frame for each shot has presented. However, a single key frame is not able to provide sufficient information about the video content of a given shot, especially for shots of long duration. Moreover, important shots of small duration may have no key frames while shots of longer duration may be represented by the multiple frames with similar content. We propose a simple intuitively appealing algorithm for finding the number of key frames allocated in each shot. This algorithm may not be optimal, but it 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

$$M_i(K_i) = CAF_i(L)2^{-2(K_i-1)} \tag{4}$$

, 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:

$$CAF_i(n) = \sum_{k=1}^{n} AF(k) \tag{5}$$

, where i, k are the shot and frame index, respectively. If the summation of eq. (5) 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. (4), 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:

$$Q_i(m) = M_i(K_i(m)) \tag{6}$$

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.

3. SIMULATIONS

The proposed key frame assignment method was validated by experiment using several long video sequences, as listed in Table 1. The test sequence was 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 frames/s. The sequence was also available as DC sequence, obtained from the MPEG stream with (slightly modified) frame sizes of 44×30. The sequences include news programs, sports, and music video.

The reduced DC sequences were first extracted using the algorithm described in section 2. Next, the shot boundaries were detected using the method from section 2. Fig. 4 illustrates the plot of $AF(k)$ vs. k for TV news sequence.

From the $AF(k)$ curve, the video sequence was

Table 1. Video sequences used in experiments

Video sequences	No. of frames	Bit rate	min:sec
TV news ("News.mpg")	10,000	1.3 Mbps	5:33
music video ("Music.mpg")	12,541	1.4 Mbps	6:58
sports ("Soccer.mpg")	20,000	1.3 Mbps	11:07

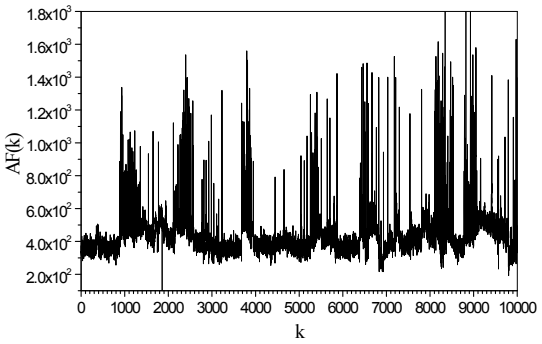

Fig. 4. Plot of $AF(k)$ vs. k for TV news sequence.

Table 2. Results of video segmentation

Video sequences	No. of frames	No. of shots T	$K_T(= T\times 1.5)$
TV news	10,000	60	90
Music video	12,391	68	102
Sports	20,000	77	116

determined to consist of 60 shots. Table 2 lists the results of the video segmentation for each sequence.

After video segmentation, key frame allocation is then performed for the individual shots obtained as described in section 2.2. In the experiments, the only parameter set was the maximal number of key frames. Table 3 shows the key frame allocation 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 in order to sufficiently describe the visual content of the shot. K_T can be adjusted by the user according to a pictorial summary and storage capacity.

we know that several previous methods which are based on threshold [9,10] are very hard to estimate the total number of key frames for the entire sequence. Therefore, this method does not provide the controllability of the total key frame number according to the capacity of storage media. Although thresholds can be updated and made shot-adaptive using statistical measures, they are essentially chosen subjectively yielding largely unpredictable results. Most likely, in a practical

Table 3. Results of proposed key frame assignment algorithm

Video sequences	$L,\ M_i(K_i(0)),\ K_i$							
TV news (60 shots)	Shot index i	1	2	3	⋯	58	59	60
	L	880	477	186	⋯	116	36	54
	$M_i(K_i(0))$	321288	276218	90656	⋯	37524	12146	24406
	K_i	4	3	2	⋯	1	1	1
Music video (68 shots)	Shot index i	1	2	3	⋯	66	67	68
	L	393	287	283	⋯	135	470	155
	$M_i(K_i(0))$	218148	127506	121744	⋯	54934	198256	58460
	K_i	3	2	2	⋯	1	3	1
Sports (77 shots)	Shot index i	1	2	3	⋯	75	76	77
	L	303	68	139	⋯	44	1164	67
	$M_i(K_i(0))$	99370	31863	64372	⋯	14916	364450	18634
	K_i	2	1	1	⋯	1	4	1

storage system a limit will exist on the number or rate of key frames to be stored because of storage limitations. To make matters worse, disadvantage of threshold-based method is that it needs (# of frames - # of shots - 1) comparisons in order to find target K_T for given threshold. For example, if a video sequence is composed of 10000 frames and it is segmented into 60 shots, 9939 comparisons are performed to find desired K_T . To investigate variation of number of key frames for content and length of shot, we plot initial value of content function, number of frames in shot, and number of key frames for shot index. In Fig. 5, the proposed algorithm produced a good compaction performance consistent with the duration and the content of each shot. Fig. 5(a) illustrates the number of assigned key frames in each shot when K_T was set at 1.5 times the number of shots. However, K_T can be adjusted by the user according to a pictorial summary. Figs. 5(b) and (c) illustrate the plot of the number of frames, and initial value of content function vs. shot index, respectively.

Unlike temporal video segmentation, an objective performance analysis method is hard to define for the assessment of a key frame assignment algorithm. Accordingly, the overall performance was evaluated by the temporal compaction achieved according to the content of each shot. Fig. 5 shows that the proposed algorithm produces a good assignment performance consistent with the

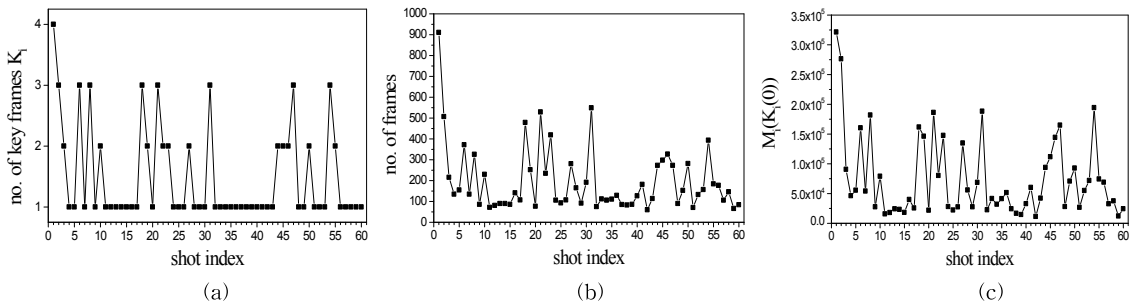

Fig. 5. Results of key frame assignment. (a) Number of key frames K_T vs. shot index ($K_T = 1.5 \times 60, g = 1$), (b) Number of frames L vs. shot index, (c) Initial value of content function $M_i(K_i(0))$ vs. shot index.

content and the duration of each shot.

4. CONCLUSIONS

Recently, 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. Moreover, there is an essential need to automatically extract key information from images and videos for the purpose of indexing, fast and easy retrieval, and scene analysis. Therefore, key frame extraction provides a powerful tool for video content summarization and visualization.

In this paper, we proposed a new and simple key frame assignment algorithm applicable to various video-indexing schemes, such as content-based browsing and retrieval from video archives. The proposed algorithm can operate directly on Motion-JPEG or MPEG compressed video and is independent of any subjective thresholds or manually set parameters. Experimental results confirmed the validity and usefulness of the proposed method. In addition, the proposed key frame allocation framework can provide a sufficient platform for many multimedia applications, efficient management of large video database, access to video archives, and the automatic creation of video clip.

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 Circuit and Systems for Video Technology,* 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," *Proceedings of IEEE ICSP'04*, Vol.1, pp. 741-744, 2004.

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

[6] Kin-Wai Sze, "A New Key Frame Representation for Video Segment Retrieval," *IEEE Trans. on Circuit and Systems for Video Technology*, 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," *Proceedings of ISCAS*, pp. II-57~II-60, 2004.

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

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

[10] Fangxia Shi and Xiaojun Guo, "Keyframe Extraction Based on K-meas Results to Adjacent DC Images Similarity," *Proceedings 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 Circuit and Systems for Video Technology*, Vol.9, No.7, pp. 1100-1114, 1999.

[12] K.W. Kim and S.G. Kwon, "Shot Motion Classification Using Partial Decoding of INTRA Picture in Compressed Video," *Journal of Korea Multimedia Society*, Vol.14, No.7, pp. 858-865, 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.

Jae-Seung Lee

He received the B.S. and M.S degrees, Electronics Engineering, Kyungpook National University, Korea in 1999 and 2001 respectively. He is currently a engineer in Korea Aerospace Research Institute. His research interests include image processing, embeded system and realtime operating system.

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.