GLOBAL MOTION COMPENSATED KEY FRAME EXTRACTION FROM COMPRESSED VIDEOS

Haoran Yi, Deepu Rajan and Liang-Tien Chia

School of Computer Engineering
Nanyang Technological University, Singapore 639798
{pg03763623, asdrajan, asltchia}@ntu.edu.sg

ABSTRACT

A key frame extraction approach based on change detection of DC images extracted from compressed video is proposed in this paper. We define a simple pixel change map that captures additional information in a frame with respect to its adjacent frames. Since global motion contributes to pixel changes, falsely indicating the presence of key frames, it is compensated by adaptively filtering the pixel change map using a modified version of the least mean square (LMS) algorithm. The prediction errors thus obtained are used to subsequently select the key frames. The key frames are selected so that the cumulative prediction error is partitioned into equal amounts in each segment. The entire procedure is computationally simple and flexible. Experimental results illustrate the good performance of the proposed algorithm.

1. INTRODUCTION

The increased availability and usage of digital videos has created a need for automated video content analysis and multimedia database management techniques. Key frames provide an abridged representation of the original video sequence and can be used for video indexing, retrieval, browsing and summarization [1]. Most of the previous approaches for video key frame extraction are based on color features (see [2] for a detailed review). The general approach was to partition the video sequence into shots and to select one or more representative frames from the shot as key frames. A naive approach, but effective only for low motion videos, is to choose the first frame of a shot as the key frame. A more robust approach was proposed in [2], in which they cluster the colour histograms of the frames and choose the key frames nearest to the centroids of the clusters. The optimal number of key frames are determined by a cluster validity analysis. Motion based approaches to key frame extraction include [3] that uses optical flow, while [1] utilizes the MPEG-7 motion activity descriptor as a measure of the summarizability of the video sequence. Lee and Kim describe a method to select key frames based on temporal variation of the mean intensity in a frame to minimize a distortion function [4].

In this paper, we present an adaptive key frame extraction method that includes compensation of camera motion for compressed videos. To process the video directly in compressed domain not only offers savings in computational resources but also provides additional information like motion vector to be utilized for global motion compensation. The motivation for global motion compensation is justified as follows. Large changes in pixel intensities between adjacent frames indicate a potential candidate for key frame. However, even a smooth motion of the camera can result in such large changes, especially in textured regions, although there is no significant change in the content and appearance of the scene. This results in redundant detection of key frames. Hence, there is a need to undo the effect of camera motion. At the same time, an irregular global motion component will indeed translate into large pixel changes pointing towards the existence of potentially valid key frames. Our method starts with computing a simple *pixel change map (PCM)* between adjacent frames. The key frame algorithm involves two passes over the PCM. The first pass is an adaptive filtering of the PCM that enables the prediction errors to be used as a cue for key frame extraction. The second pass involves the actual selection of the key frames.

2. KEY FRAME EXTRACTION

The proposed key frame extraction algorithm consists of 3 components: (1) computing pixel change maps, (2) camera motion compensation by adaptive filtering and (3) key frame selection.

2.1. Pixel Change Map

The objective of the *pixel change map (PCM)* is to capture the presence of additional information vis-a-vis the current frame. This additional information could be due to the presence of a shot boundary or due to changes in the foreground and/or background. If the current frame occurs between two shot boundaries, it is similar to its neighboring frames and little additional information is contained in it. On the other hand, changes in background/foreground result

in large changes in pixel values indicating significant additional information. Thus, the PCM is obtained by thresholding the difference between the current frame and the adjacent frames. However, we would like to achieve this in the *compressed domain*. Hence, we use DC images to compute the PCM instead of fully decoding the frames. The DC images for I and P frames are extracted using the method in [5]. The DC images are spatially reduced versions of the original frames and are less prone to noise and illumination changes. For the current DC image p_i corresponding to frame i, we determine $DI_i = |p_i - p_{i-1}| + |p_{i+1} - p_i|$. For each pixel in frame i, if DI_i is greater than a threshold, the pixel is declared as "changed pixel" and the corresponding location in PCM_i (PCM for frame i) is set to one; otherwise it is set to zero. The comparison of DI_i with a threshold is simply to undo the effect of any noise associated with the camera or the partial decoding process of the compressed videos and has no critical bearing on the algorithm for key frame extraction.

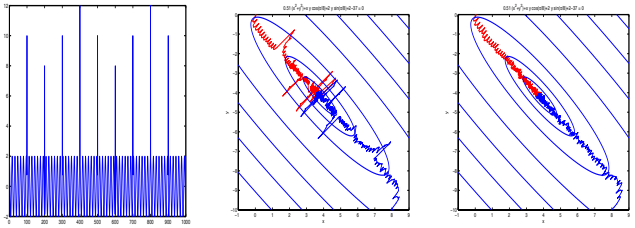
2.2. Global Motion Compensation By Adaptive Filter

As mentioned earlier, even a little camera motion can result in large pixel changes if the scene is textured. However, such pixel changes are undesirable because the subsequent key frame extraction algorithm will be falsely triggered at these points. Hence, there is a need to compensate for camera motion. While there are several methods that determine global motion which can be used to warp the frame using estimated motion parameters, there are shortcomings to this approach. In the context of compressed domain processing, most global motion estimation approaches use motion vectors, which are codec dependent and may not be accurate. Moreover, the warping of the frames using estimated motion parameters (e.g. using an affine model) requires significant computation, e.g., for each $M \times N$ frame, the computation for warping one frame using bilinear interpolation requires $4 \times M \times N$ multiplications and $4 \times M \times N$ additions.

Based on the observation that camera motion always lasts for a few frames and that the camera motion is much smoother than object motion, we ascertain that the pixel changes due to camera motion will have a strong local correlation. This allows us to model the pixel changes as an auto-regressive (AR) process

$$c(n) = \sum_{i=1}^{h} w(n-i) \cdot c(n-i), \qquad (1)$$

where $c(n)$ is the pixel change at location n (assuming lexicographic ordering of the frame), w is the coefficient and h is number of the taps of the AR model. The errors from the local regression with the AR model are deemed as the true pixel changes, i.e., additional information carried by each frame and used as input for the algorithm described in Section 2.3 to extract the key frames. It is these prediction errors that serve as a cue to the correct extraction of key frames in that large errors occur due to the difficulty in predicting the next frame, which in turn is attributed to significant motion.



(a) (b) (c)

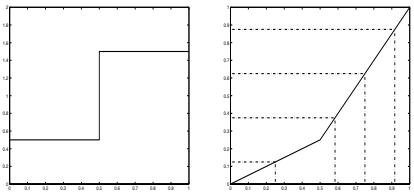
Fig. 1. (a) The simulated pixel change sequence. Convergence of weights using (b) LMS algorithm and (c) SLMS algorithm.

We use a modified version of the Least Mean Squares (LMS) algorithm to determine the AR parameters[6]. The reason for modification is illustrated by the following scenario. Consider a shot boundary. The pixel changes corresponding to a shot boundary will be much higher relative to other frames. This will result in a large error in prediction. Note that the large error is *useful* for key frame selection. However, it will also update the AR coefficients drastically causing a drift in the weights. Hence we modify the LMS algorithm to the extent that the weights are updated only if the error is less than a threshold. We call this process the $Switched - LMS (SLMS)$ algorithm.

Consider an AR process with input $x(n) = \sin(\frac{n\pi}{8}) + \sigma(n)$, the desired signal (the pixel change sequence in this case) $d(n) = 2\cos(\frac{n\pi}{8}) + T \cdot \delta(mod(T, 100))$, where $\sigma(n)$ is the white Gaussian noise process with variance equal to 0.1 and $T \cdot \delta(mod(T, 100)$ simulates (periodic) shot changes. Figure 1 (a) shows the simulated random process $d(n)$ and Figures 1(b) and (c) show the convergence of the adaptive weights for LMS and SLMS algorithms, respectively. Clearly, the SLMS converges more smoothly and faster than the LMS algorithm.

2.3. Key Frame Selection

As we have seen in Section 2.2, modeling the PCM as an AR process yields prediction errors, PCM_e, that captures the additional information at a particular frame. To this extent, the total amount of information contained in an entire video sequence is deemed to be captured by the accumulation of the PCM_es for the sequence. Note that the PCM values for the first frame are set to 1, the AR coefficients are initiallized to 0 and, therefore, the prediction errors PCM_e are equal to 1 for the first frame. Given the total information in a sequence (corresponding to accumulated PCM_es), the objective is to partition the sequence with key frames



(a) (b)

Fig. 2. (a) Prediction error in pixel change, PCM_e, per frame (synthetic data), (b) Cumulative PCM_e with location of extracted key frames.

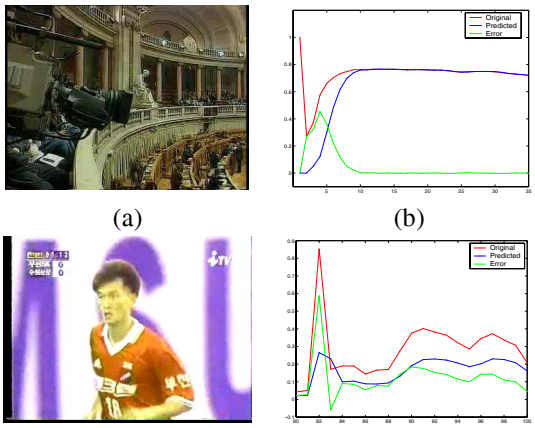
such that each segment contains equal amount of information. Hence the key frames act as pegs in the cumulative PCM_es such that their locations force equal information in every interval.

Let PCM_e^i be the prediction error of the pixel change map at frame i and N be the number of key frames to be extracted. We compute the cumulative PCM_e for the sequence as $CPCM_e = \sum_{j=1}^{M} PCM_e^j$, where M is the number of frames. Next, we traverse the cumulative PCM_e marking off points $l = \frac{l}{N} - \frac{1}{2N}$, $l = 1, \cdots, N$. The abscissa of each of these points marks the location of the key frame. Note that this method does not need any information about the location of shot boundaries. Moreover, a single pass through the cumulative PCM_e is enough to select the location of the key frames.

Figure 2 illustrates the proposed key frame extraction on synthetic data. From the prediction error of pixel change in Figure 2 (a), we see that the data has two distinct segments: the PCM_e per frame is 0.5 in the first half while it is 1.5 in the second half. Suppose 4 key frames are to be extracted. The cumulative PCM_e, shown in Fig 2 (b) is marked off at 1/8, 3/8, 5/8, 7/8 and the corresponding abscissa that locate the key frames are extracted at 1/4, 7/12, 3/4 and 11/12. The PCM_e in the second half segment is 3 times that in the first half segment. Thus, the number of key frames extracted from the former should be three times that of the key frames extracted from the latter. The proposed method has successfully extracted one key frame and three key frames from the first and second half segments, respectively. This is consistent with our assertion that larger prediction errors indicate the presence of more key frames. Note that no information about shot boundaries was used to locate the key frames.

3. EXPERIMENTAL RESULTS

In this section, we describe two sets of experiments. The first set evaluates the effectiveness of the S-LMS adaptive filtering algorithm to compensate for the effect of camera motion on pixel changes and to generate perceptually consistent prediction errors. The second set evaluates the proposed key frame extraction algorithm.



(a) (b)

(c) (d)

Fig. 3. Frame from video clip with (a) smooth and (c) irregular camera motion. (b) and (d) Original, predicted and prediction errors of pixel changes for video clip of (a) and (c), respectively(Horizontal axis shows the frame number.)

3.1. Adaptive Filtering for Global Motion Compensation

We choose two sequences to illustrate the adaptive filtering on the PCMs. The first sequence contains only smooth camera motion consisting of a pan from left to right. An example frame from the sequence is shown in Figure 3(a). The actual pixel changes (red), predicted pixel changes (blue), and the prediction error (green) are shown in Figure 3(b). We see that the error drops very quickly after a few frames and remains so till the end of the shot. Since the error is low, we conclude that there is no significant activity and hence the number of key frames extracted should be less. The second sequence consists of the camera tracking a soccer player on the field. Since the player moves quite irregularly, the motion of the camera is not smooth and the values of the pixels change rapidly. An example frame for the video clip is shown in Figure 3 (c). The actual pixel changes (red), predicted pixel changes (blue), and the prediction error (green) are shown in Figure 3(d). The jerky motion of the camera results in little local correlation in the PCM and hence a higher prediction error. However, such high errors are desirable for key frame selection. The content in this video interval is of high motion and carries more information than still or smooth camera motion video intervals. Therefore, more emphasis should be put on such intervals for key frame selection, i.e., more key frames should be selected.

3.2. Key Frame Extraction

We consider a sequence from the MPEG-7 test set which consists of four shots in which the first two shots track a group of people walking, the third shot shows a person answering questions while the fourth shot is that of an anchor person. The 4 keyframes extracted from this sequence are

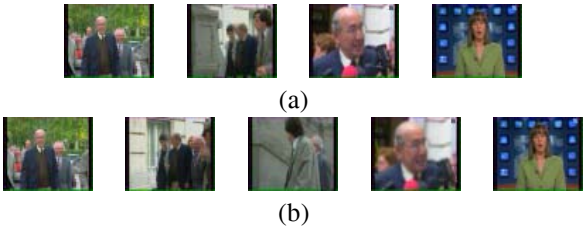


Fig. 4. DC images of (a) 4 keyframes(Frame #1,#67, #139, #374) and (b) 5 keyframes(Frame #1, #45, #96 , #192,#374) from an MPEG-7 test sequence.

shown in Figure 4(a) (as DC images) where each key frame is extracted from each shot. Note, however, that our method does not presume the presence of a shot boundary or use any information about the location of shot boundaries. Figure 4(b) shows 5 key frames extracted from the same sequence. The additional key frame is that of a turning person; thus, the key frames are essentially able to capture the important events in the sequence.

Next, we compare the proposed algorithm with two other methods. The first one (abbreviated CIK) selects key frames at constant time intervals and the second (abbreviated CTK) is based on clustering the colour histograms of the frames so that the frame closest to the centroid of the cluster is chosen as the key frame. The performance of each method is compared by the average histogram error defined as $H_{err} = \frac{1}{N}\sum_{i=1}^{N} \min_{j\in K}(dist(H_i, H_j))$, where H_i is the histogram for frame i, N is the total number of frames, K is the set of key frames and $dist$ is the distance function between two histograms chosen as the summation of the absolute difference between the two histograms. Table 1 shows the average histogram errors on 7 video sequences each containing 300 frames. 4 key frames are selected. The proposed method outperforms CIK. Although *CTK* is shown to achieve the lowest average histogram error, it does not take temporal information into consideration leading to incorrect selection of key frames from a perceptual point of view. This is illustrated in Figure 5(a) where 2 key frames are extracted from the first court-view shot, but none from the second court-view shot. However, the proposed method exhibits proper evolution of the video content as shown in Figure 5(b). The clustering method is also much more computationally intensive than the proposed method.

4. CONCLUSION

In this paper, we propose a novel approach for key frame extraction for compressed videos. The total information of the video is represented by accumulation of prediction errors of pixel change maps. These errors are obtained after adaptive filtering of the PCM to compensate for the effect of global motion.The prediction errors are then used as a cue for key frame extraction. The key frames are selected such that the video is partitioned into segments with

Fig. 5. DC images of key frames extracted by (a) CTK method(Frame #37, #112, #211,#256) and (b) the proposed method(Frame #106, #214, #241, #277).

Videos	CIK	CTK	Proposed
Basketball 1	0.045	0.036	0.038
Basketball 2	0.082	0.058	0.071
News 1	0.087	0.058	0.067
Soccer 1	0.078	0.045	0.070
Soccer 2	0.088	0.053	0.075
TV 1	0.027	0.015	0.025
TV 2	0.031	0.020	0.023

Table 1. Average Histogram errors for CIK, CTK and the proposed method.

equal amount of prediction error of pixel changes. All the computation is carried out with DC images extracted from compressed video directly. The proposed method is computationally simple and adaptive and can rapidly generate key frame based video summary of any desired length. The experimental results and comparison with other methods show that the proposed method offers advantages.

5. REFERENCES

[1] A. Divakaran, R. Regunathan, and K. A. Peker, "Video summarization using descriptors of motion activity: A motion activity based approach to key-frame extraction from video shots," *Journal of Electronic Imaging*, vol. 10, pp. 909–916, Oct. 2001.

[2] A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1280–1289, Dec. 1999.

[3] W. Wolf, "Key frame selection by motion analysis," in *Proceedings of the ICASSP*, Atlanta, USA, Apr. 1996, pp. 1228–1231.

[4] H. C. Lee and S. D. Kim, "Rate-driven key frame selection using temporal variation of visual content," *IEE Electronics Letters*, vol. 38, no. 5, pp. 217–218, Feb. 2002.

[5] B. L. Yeo and B. Liu, "On the extraction of DC sequence from MPEG compressed video," in *Proceedings of the IEEE ICIP*, Washington, DC, USA, 1995.

[6] S. Haykin, *Adaptive Filter Theory*, NJ: PrenticeHall, 2002.