

Screen image sequence compression method utilizing adaptive block size coding and hierarchical GOP structure

WU Xing(武星)¹, MEI Liang(梅亮)², XI Qi(袭奇)³, ZHANG Shen-sheng(张申生)³, CHEN Yan-wei(陈延伟)⁴

1. School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;
2. Department of Electrical Engineering and Computer Science, University of Michigan, Michigan 48109, USA;
3. Joint Institute of University of Michigan–Shanghai Jiao Tong University, Shanghai Jiao Tong University, Shanghai 200240, China;
4. College of Information Science and Engineering, Ritsumeikan University, Shiga 5258577, Japan

© Central South University Press and Springer-Verlag Berlin Heidelberg 2010

Abstract: To compress screen image sequence in real-time remote and interactive applications, a novel compression method is proposed. The proposed method is named as CABHG. CABHG employs hybrid coding schemes that consist of intra-frame and inter-frame coding modes. The intra-frame coding is a rate-distortion optimized adaptive block size that can be also used for the compression of a single screen image. The inter-frame coding utilizes hierarchical group of pictures (GOP) structure to improve system performance during random accesses and fast-backward scans. Experimental results demonstrate that the proposed CABHG method has approximately 47%–48% higher compression ratio and 46%–53% lower CPU utilization than professional screen image sequence codecs such as TechSmith Ensharpen codec and Sorenson 3 codec. Compared with general video codecs such as H.264 codec, XviD MPEG-4 codec and Apple's Animation codec, CABHG also shows 87%–88% higher compression ratio and 64%–81% lower CPU utilization than these general video codecs.

Key words: screen image sequence; compression; adaptive block size; hierarchical GOP structure; intra-frame coding; inter-frame coding

1 Introduction

Screen images and screen image sequences need to be compressed and transmitted in many real-time remote and interactive applications, such as E-learning, distance collaboration and remote monitoring and control. Although there are some professional screen image sequence codecs such as TechSmith EnSharpen codec [1] and Sorenson 3 codec [2], the bottleneck of these codecs is a relatively high CPU utilization and unacceptable compression performance for real-time applications.

Studies on a single screen image compression have been made in recent years without exploiting the temporal redundancy of screen image sequences. ZHANG et al [3] proposed an image compression method based on image adaptive segmentation and adaptive quantification, which can be used for single screen image compression. In this method, the smallest block size is 8×8 , which is not small enough to produce satisfactory coding performance in image areas

containing high-frequency components, such as texts. AGU-MHV [4] is a bit-plane-based efficient coding of discrete cosine transform (DCT) coefficients, which provides better performance than JPEG2000 [5] and outperforms the baseline JPEG [6]. However, the inadequacy of the AGU-MHV coder is a high computational complexity in both partitioning of an image into blocks and bit-plane coding of transform coefficients. LIU et al [7] proposed a screen image coding algorithm with quality-biased rate allocations for block classification, in which image segmentation [8] plays an important role in image compression.

There is also some research work focused on special screen image sequence compression. DESHPANDE and HWANG [9] proposed a method to compress the screen image sequence using JBIG encoding method. It works well for the screen image sequence of handwritten text, but is not well suitable for handling general screen image sequences. YIN et al [10] described a screen image compression algorithm for E-learning system, but the screen share module divided screen images into four types

Foundation item: Project(60873230) supported by the National Natural Science Foundation of China

Received date: 2009–09–26; **Accepted date:** 2010–01–22

Corresponding author: WU Xing, PhD; Tel: +86–13564333050; E-mail: xingwuvip@gmail.com

of regions, which increases the computational complexity.

It is indispensable to propose a compression method that achieves better compression performance than existing codecs. The proposed intra-frame coding algorithm in CABHG is an advanced DCT based variable block size (from 16×16 to 4×4) image compression method. Based on previously introduced image resizing methods [11] operating in transform domain, an improved block combination method is introduced. Furthermore, the rate-distortion cost function of H.264/AVC [12–13] is utilized for block partition. For the inter-frame coding algorithm in CABHG, an improved hierarchical group of pictures (GOP) structure [14] is presented to improve system performance during random accesses and fast-backward scans.

2 Overview of proposed CABHG

2.1 Characteristics of screen images and screen image sequences

For the purpose of high compression performance, the characteristics of screen images and screen image sequences need to be considered. Different from continuous tone natural images with abundant colors, screen images have less colors and the main part of the screen images usually consists of simple text and graphical user interface (GUI) icons, though sometimes complex pictures. Furthermore, screen image sequence is different from natural video for its discontinuous change [15]. Due to being in general stationary in the temporal axis, screen image sequences can be efficiently represented at a lower frame rate compared with natural video. That is, 8–15 frames per second will meet the requirement in most cases. Although screen image sequences have limited motion, there is still relevance between adjacent frames. Most of the time, two adjacent frames are identical or vast number of pixels in

successive frames do not change.

Being concerned with real-time compression and transmission of screen image sequences, three features are essential to an expected compression algorithm: a very low complexity, a good image quality after decoding, and a high compression ratio.

2.2 Flowchart of proposed CABHG

Based on the above-mentioned characteristics and requirements, the flowchart of the CABHG system is presented in Fig.1. In the CABHG method, current frame of a screen image sequence is first compared with the previous frame to determine whether the current frame is copied or not. When the two frames are identical, only the header of the current frame is coded, which notifies the decoder to copy the previous frame. If the two adjacent frames are different, the selection between inter-frame coding and intra-frame coding should be considered. The proposed intra-frame coding is a rate-distortion optimized adaptive block size image coding method that utilizes the inner-block and inter-block information to guide the decision of block size and partition scheme. While in the inter-frame coding algorithm, two modes (I-mode and P-mode) are switched considering the motion characteristics of the screen image sequence. The decoding process is the reverse flow of the encoding process and is not discussed in this work.

3 Intra-frame coding

In H.264/AVC intra-frame coding, the intra-prediction scheme is introduced for encoding I-frames. This scheme can reduce the bit size of an I-frame and maintain a high quality by enabling the successive prediction of smaller blocks of pixels within each macro-block in a frame. It is the rate-distortion optimization that is employed to select the optimal coding

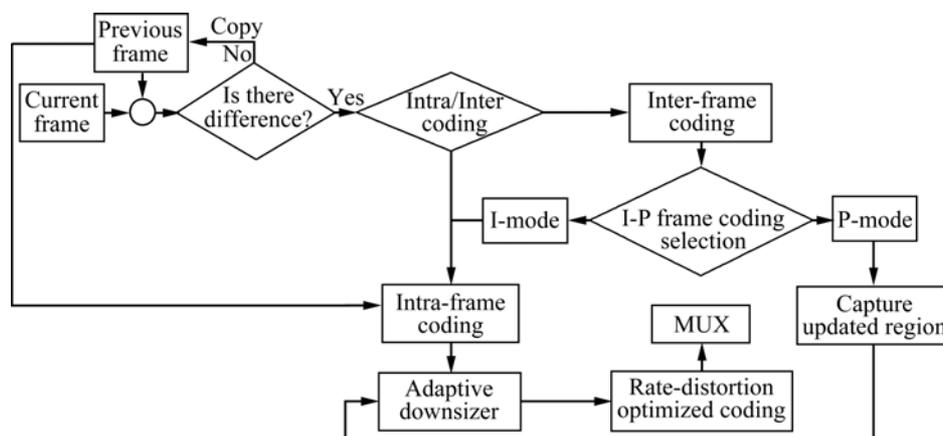


Fig.1 Flowchart of proposed CABHG method

mode which achieves the minimum rate-distortion cost based on the distortion model using Lagrangian coefficients. The encoder will search all possible modes exhaustively to encode the target block for the best modes, which will increase computational complexities. This case is the same as that in the AGU-MHV method. These methods with high computational complexity are not suitable for real-time applications.

It is necessary to propose a new rate-distortion optimized adaptive block size image coding method utilizing transform domain resizing with a low computational complexity. The proposed algorithm first partitions the image into blocks of median size, and then searches the possibility of block combination and partition in accordance with the characteristics of different regions of one screen image. After the block size optimization, the image is encoded with adaptive block sizes. The proposed adaptive block coding method based on the standard JPEG baseline sequential coding method, and several additional components are introduced to the coding process. The block diagram of the intra-frame coding process is shown in Fig.2.

The scheme starts from partitioning the original image into 16×16 macro-block (MB). Then, the MB is converted from 4:4:4 (RGB) to 4:2:0 (YcbCr) (Ycb and Cr are the luma component, the blue-difference and red-difference chroma components, respectively) representation. The chrominance channels are downsampled, forward discrete cosine transformed and encoded as defined in the standard JPEG. Because human eyes are not sensitive to chrominance signals, adaptive block size optimization is not performed for

image blocks of those channels. Y channel components are not directly encoded as defined in the JPEG standard. An adaptive downsizing algorithm is applied to the neighboring blocks in a 16×16 MB, seeking to combine similar blocks vertically and/or horizontally. For blocks that do not meet the criteria for combination, the rate-distortion based criterion is utilized to check whether it should be segmented into 4×4 sub-blocks. If this is the case, the block is divided into four 4×4 sub-blocks, a 4-point FDCT is performed, and then the block is encoded separately. Otherwise, the 8-point FDCT is performed and the entire block is encoded just as that suggested in standard JPEG.

After proper partition, coefficients are quantized, zigzag reordered, run-length encoded and mapped into the bitstream using Huffman tables. For those 8×8 blocks, the quantization tables and Huffman code tables suggested by the JPEG are used. For those 4×4 blocks, quantization tables are determined according to the idea proposed by LIM et al [16]. Since Huffman encoding is based on the probability of the occurrence of symbols, Huffman code tables used for coding 8×8 blocks can be used for 4×4 blocks by adjusting the possibility distribution of DCT coefficients in 4×4 blocks.

3.1 DCT and block similarity

With the above mentioned two-dimensional (2D) FDCT, a spatial block $f=f_{ij}$ is transformed to a frequency block $F_{u,v}$ as follows:

$$F_{u,v} = C_{u,v} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} f_{i,j} B_{i,j}^{u,v}, 0 \leq u, v \leq M-1 \quad (1)$$

where

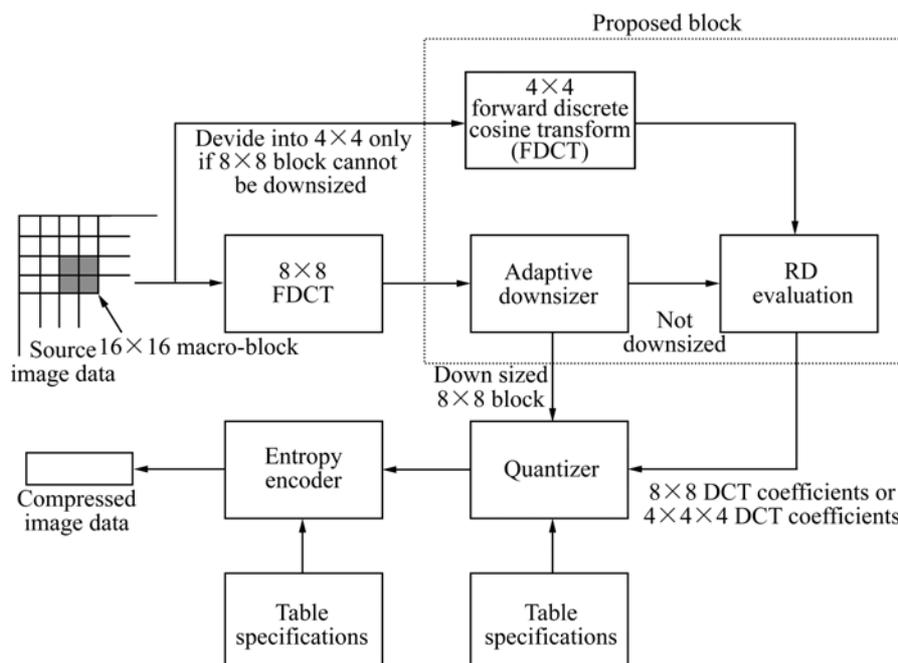


Fig.2 Block diagram of proposed intra-frame coding process

$$C_{u,v} = \begin{cases} 1/M, & u = v = 0 \\ \sqrt{2}/M, & u = 0 \text{ or } v = 0 \\ 2/M, & u \neq 0 \text{ and } v \neq 0 \end{cases} \quad (2)$$

$$B_{i,j}^{u,v} = \cos\left[\frac{(2i+1)u\pi}{2M}\right] \cos\left[\frac{(2j+1)v\pi}{2M}\right] \quad (3)$$

After the FDCT is performed, the upper left corner of the coefficient matrix contains the DC component $F_{0,0}$ that determines the mean of f . The horizontal frequency components F_H , vertical frequency components F_V and remaining high-frequency components F_{AC} can be expressed as

$$F_H = [F_{1,0}, F_{2,0}, \dots, F_{M-1,0}]^T \quad (4)$$

$$F_V = [F_{0,1}, F_{0,2}, \dots, F_{0,M-1}] \quad (5)$$

$$F_{AC} = \{F_{u,v}, 1 \leq u, v \leq M-1\}_{M-1, M-1} \quad (6)$$

The inverse DCT is defined as follows:

$$\hat{f}_{i,j} = \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} C_{u,v} \hat{F}_{u,v} B_{i,j}^{u,v}, 0 \leq u, v \leq M-1 \quad (7)$$

where $\hat{f}_{i,j}$ is the decoded block, and $\hat{F}_{u,v}$ is the quantized DCT coefficient of $F_{u,v}$ under a quantization level Q .

Based on the boundary difference [17–18], block similarity (S) between benchmark block B and comparative block C in CABHG is proposed and defined as

$$S = \frac{1}{\hat{f}_{i,0}^B - \hat{f}_{i,M-1}^C} \quad (8)$$

With Eq.(7), parts of Eq.(8) can be expressed as

$$\hat{f}_{i,0}^B = \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} C_{u,v} \hat{F}_{u,v}^B B_{i,0}^{u,v} \quad (9)$$

$$\hat{f}_{i,M-1}^C = \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} C_{u,v} \hat{F}_{u,v}^C B_{i,N-1}^{u,v} \quad (10)$$

where $\hat{f}_{i,0}^B$ means the first column of the benchmark block and $\hat{f}_{i,M-1}^C$ means the last column of the comparative block.

Block similarity between two adjacent blocks is determined by the DC component $F_{0,0}$, horizontal frequency component F_H , vertical frequency component F_V and remaining high-frequency component F_{AC} of them. It is proposed that the approximated value of the block similarity between two adjacent blocks can be simplified as

$$S = 1/(|\hat{F}_{0,0}^B - \hat{F}_{0,0}^C| + |\hat{F}_{0,1}^B + \hat{F}_{0,1}^C| + |\hat{F}_{0,2}^B - \hat{F}_{0,2}^C| + |\hat{F}_{1,0}^B - \hat{F}_{1,0}^C| + |\hat{F}_{2,0}^B - \hat{F}_{2,0}^C|) \quad (11)$$

3.2 Block pattern decision

With the proposed block similarity definition in

previous subsection, the block pattern is determined for adaptive downsizing. Downsizing is defined as the block combination in the DCT domain. The judgment of the combination possibility of two adjacent blocks depends on two criteria. In the first place, two blocks should be similar enough. Secondly, high-frequency coefficients of the two blocks can be neglected.

The similarity between two neighboring 8×8 blocks is testified with Eq.(11). If the similarity value S of two adjacent blocks exceeds a fixed threshold B_{TH} , the first criterion holds true and the judgment for the second criterion should be carried out. Notice that though threshold B_{TH} is fixed during one compression process, experimental results show that its optimal value varies with different compression ratios, so this value should be adjusted according to the quantization level.

The high-frequency coefficients of the two adjacent blocks will determine whether the high-frequency information can be neglected or not. For horizontal downsizing, the high-frequency information can be ignored if the right-half DCT coefficients are equal to zeros after quantization. The judgment for vertical downsizing is performed in a similar way except that the lower-half DCT coefficients of two vertical blocks are evaluated. As the downsizing scheme will truncate the high-frequency coefficients of the two blocks, blocks that have large high-frequency coefficients are not suitable for combination because doing so will lead to a high degree of distortion.

If the two criteria are satisfied, the two neighboring blocks can be combined together and the next two blocks will then be analyzed. Fig.3 illustrates the combination possibilities for a 16×16 MB that contains four original 8×8 blocks. First of all, the horizontal combination is performed, from the right to the left. If the two blocks meet the combination criteria, the right block is combined into the left block and the left block holds the results of the combined coefficients. Then, vertical combination is considered. At this stage, the blocks may contain the horizontally combined coefficients, but they are not treated differently from the original 8×8 blocks. The bottom block is combined into the upper block and the upper block holds the results of the combined coefficients. As shown in Fig.3, after the horizontal combination stage, there are four possibilities for vertical combinations to continue. After the vertical combination stage, there are eight final possible combined results for a 16×16 MB, representing 10 block patterns in Fig.4, which need to be recorded in the final bitstream.

The best scenario is that both rows of 8×8 blocks are horizontally combined and the resulting two 8×8 blocks are then vertically combined. In this situation, the

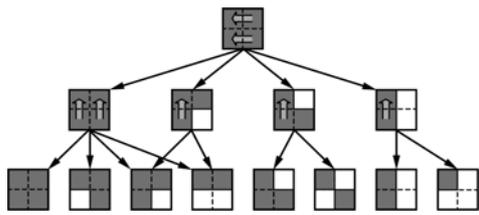


Fig.3 Combination possibility for adjacent blocks in 16×16 MB

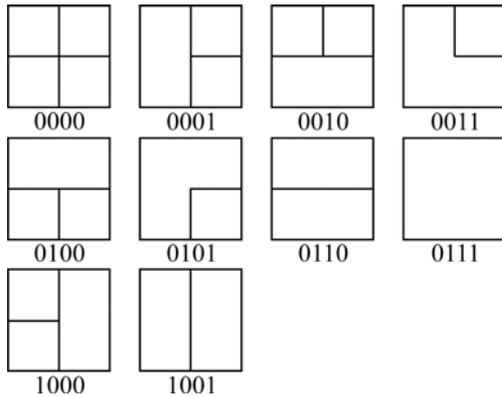


Fig.4 Ten possible block patterns in proposed combination scheme

information of four neighboring 8×8 blocks can be stored into one single 8×8 block of coefficients. The worst scenario is that the four 8×8 blocks in the 16×16 MB cannot be combined either horizontally or vertically.

3.3 Image resizing in DCT domains

The proposed downsizing and upsizing operation is a modified fast image resizing scheme performed in the compressed domain [19–20]. Originally, by combining the low-frequency information of four adjacent blocks into a single block or expanding the DCT coefficients of one block into four blocks, a ratio of two fast image scaling is achieved. This scheme is modified for the proposed CABHG: combining the DCT coefficients of two neighboring 8×8 blocks into one block, or reconstructing two blocks from the downsampled one.

Let B_1 and B_2 denote the DCT coefficients of two horizontally consecutive 8×8 blocks. The downsizing method begins with a truncation of the high-frequency coefficients of B_1 and B_2 (that is, 4 columns per block), and two half-blocks B_1 and B_2 of size 8×4 are obtained. Next, those two half-blocks are transformed with an 8×4 2D inverse DCT into the spatial domain and these two 8×4 spatial half-blocks are concatenated to constitute the new block. Finally, the combined spatial coefficients of size 8×8 are transformed back into the DCT domain with 8-point 2D DCT, and the final downsized block B is obtained. The block downsizing operation in CABHG can be written as

$$B = [\hat{B}_1 T_4 \quad \hat{B}_2 T_4] \begin{bmatrix} T_L^T \\ T_R^T \end{bmatrix} = \hat{B}_1 (T_L T_4^T)^T + \hat{B}_2 (T_R T_4^T)^T \quad (12)$$

where the 8-point FDCT operator matrix T is separated into two 8×4 matrices, T_L and T_R , denoting the first and last four columns of T , and T_4 is the 4-point inverse DCT operator matrix.

Nearly 50% of the coefficients in matrices $T_L T_4^T$ and $T_R T_4^T$ are equal to zero. By decomposing the two matrices and reordering the computation slightly, there will still be only half of the nonzero coefficients left. By defining $T_L T_4^T = C + D$ and $T_R T_4^T = C - D$, block downsizing operation in CABHG, i.e., Eq.(12), can be expressed as

$$B = \hat{B}_1 (C + D)^T + \hat{B}_2 (C - D)^T = (\hat{B}_1 + \hat{B}_2) C^T + (\hat{B}_1 - \hat{B}_2) D^T \quad (13)$$

where matrices C and D are extremely sparse with nearly 70% zero coefficients. The existence of such a large amount of zero terms in matrices C and D can effectively enhance the efficiency of the proposed downsizing scheme it is shown that the amount of computation is only 1.25 multiplications and 1.25 additions per pixel of original blocks B_1 and B_2 .

Because of the orthogonality of DCT matrices, there are also orthogonal relationships between the two matrices in Eq.(12), which can be utilized to easily derive the upsizing scheme from the downsizing scheme.

$$\begin{cases} (T_L T_4)^T (T_L T_4) = (T_R T_4)^T (T_R T_4) \\ (T_R T_4)^T (T_L T_4) = (T_L T_4)^T (T_R T_4) \end{cases} \quad (14)$$

In combination with Eq.(13), the upsizing operations in CABHG can be defined as

$$\begin{cases} \hat{B}_1 = \hat{B}(C + D) = \hat{B}C + \hat{B}D \\ \hat{B}_2 = \hat{B}(C - D) = \hat{B}C - \hat{B}D \end{cases} \quad (15)$$

Similar to downsizing, the upsizing operation can be completed with 1.25 multiplications and 1.25 additions per pixel of the upsampled blocks \hat{B}_1 and \hat{B}_2 .

3.4 Rate-distortion optimized partition

The rate-distortion cost function is utilized for the block partition decision after block combination in homogeneous regions of a frame image. During the block combination, some blocks are combined into one single block, while other blocks remain intact because they do not meet the criteria mentioned in section 3.2. Those intact blocks are often residing in areas containing high-frequency components that need further partition to improve the compression performance. Thus, it is

proposed to partition those blocks into 4×4 sub-blocks. After the partition, the rate-distortion cost is evaluated for both the 8×8 block size scheme and the 4×4 sub-block scheme. The purpose for the evaluation is to figure out whether a particular block needs further partition or not. The scheme with lower rate-distortion cost will be chosen to encode the block.

The rate-distortion cost for each 8×8 block is defined as

$$C_8 = S_{SD} + \lambda R \tag{16}$$

where R represents the number of bits used for this block; λ is Lagrange parameter; and S_{SD} is the sum of the squared difference of the reconstructed block R and the original block B . Similarly, if the block is partitioned into four 4×4 sub-blocks, the total rate-distortion cost should be

$$C_4 = \sum_{i=1}^4 (S_{SD_i} + \lambda R_i) \tag{17}$$

where R_i represents the number of bits used for compressing the i th sub-block; and S_{SD_i} represents the sum of the squared difference of the reconstructed sub-block R_i and original sub-block B_i . λ in Eqs.(16) and (17) is an empirical parameter, which changes exponentially in relation to quantization level Q_i mentioned in section 3.1.

4 Inter-frame coding

4.1 Two-mode inter-frame coding

For the inter-frame coding, the difference between screen image sequences and natural videos should be concerned. The screen changes slowly and regularly and most parts of the screen remain unchanged for a period. Two-mode inter-frame coding is introduced based on the characteristics of screen image sequences. I-mode (intra-frame coding mode) and P-mode (prediction-frame coding mode) are switched in accordance with different motions of screen images. The selection between I-mode and P-mode depends on the matching criteria as follows.

Current frame is represented with the array $curdata[m][n]$, and its previous frame is represented by $predata[m][n]$. The range of m and n depends on the screen image resolution. $rel[m][n]$ is a function of the relationship between $curdata[m][n]$ and $predata[m][n]$. It is assumed that the screen image resolution is $M \times N$, thus

$$rel[m][n] = [\text{sign}(curdata[m][n] - predata[m][n])]^2, \tag{18}$$

$$1 \leq m \leq M, 1 \leq n \leq N$$

It is obvious that the value of $rel[m][n]$ is 1 or 0. The weight of $rel[m][n]$ is

$$W = \sum_{0 \leq m}^M \sum_{0 \leq n}^N (rel[m][n]) / (MN) \tag{19}$$

where W represents the correlation between the current frame and the previous frame. The smaller the W , the stronger the correlation between the two adjacent frames. When W is smaller than a predetermined threshold, the correlation between two frames is strong enough and the current frame is coded in P-mode, otherwise it is coded in I-mode.

For screen image sequences, the inter-frame coding algorithm is proposed, and two kinds of frames are defined as follows.

(1) I-frame (intra-frame) is coded entirely in the I-mode. I-frame is coded in the following two cases: the first frame of a sequence is always coded as an I-frame; and to prevent potential error propagation and to enable random accesses, I-frame is used periodically.

(2) P-frame (prediction frame) is coded with reference to previous frames. The current frame is coded in P-mode when the correlation between the two adjacent frames is satisfied with the above mentioned conditions. The updated-region is coded as a normal image with the intra-frame coding algorithm.

The updated-region is captured by the mirror video driver in the Windows system so that the system can be quickly and efficiently notified with screen changes. The video driver also makes a direct link between the mirror video driver frame-buffer memory and the CABHG system, which can improve the system performance. With mirror video driver, CABHG can even record videos playing on the screen.

4.2 Hierarchical GOP structure

Based on previously introduced binary tree GOP in H.264/AVC, hierarchical GOP structure is proposed for CABHG. The applications of screen image transmission, such as E-learning and distance collaboration, request random accesses or fast-backward scans. These requests will ask the server to send a frame with an arbitrary distance from the current frame. With traditional frames' linear GOP structure in Fig.5, supposing all the GOPs in the bitstream have size n , the average number of frames

to be sent is $\sum_{i=1}^n i/n = (n+1)/2$.



Fig.5 Linear GOP structure of frames

The reshaped coding structure of the bitstream is a frames' binary tree structure shown in Fig.6. For convenience, the size of frames' binary tree GOP n is set

to be 2^k-1 , where k is a natural number. Thus, the average number of frames to be transmitted in this hierarchical coding scheme is

$$\frac{\sum_{i=0}^{\log_2 n-1} 2^i (i+1)}{n} = \log_2 n - 1 + \frac{n}{2} \approx \log_2 n - 1 \quad (20)$$

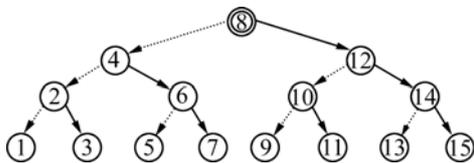


Fig.6 Binary tree GOP structure of frames

This reshaped frames' binary tree GOP reduces the number of frames to be sent from $(n+1)/2$ to $\log_2 n - 1$ and improves the system efficiency from $2/(n+1)$ to $1/(\log_2 n - 1)$. Nevertheless, the system performance can be further improved by grouping the GOPs into hierarchical structure shown in Fig.7. That is to say, the GOPs are also arranged in the binary structure rather than the linear structure. Thus, the transmission sequence for the GOPs is: GOP₈, GOP₄, GOP₂, GOP₁, GOP₃, GOP₆, GOP₅, GOP₇, GOP₁₂, GOP₁₀, GOP₉, GOP₁₁, GOP₁₄, GOP₁₃, and GOP₁₅.

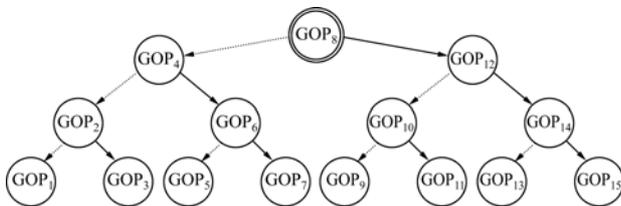


Fig.7 Binary tree structure of GOPs

The difference between Fig.6 and Fig.7 is that each node in Fig.6 is a frame, while each node in Fig.7 is a GOP. With the proposed hierarchical GOP structure, the efficiency of CABHG system in random accesses or fast-backward scans can be further improved from $1/\{(n+1)/2\}(\log_2 n - 1)$ to $1/(\log_2 n - 1)^2$.

5 Comparison and evaluation

In this section, the performance of the proposed CABHG is evaluated and compared from existing professional screen image sequence codecs such as TechSmith Ensharpen codec and Sorenson 3 codec to general video codecs such as MainConcept H.264 codec [21], XviD MPEG-4 codec [22], and Apple's Animation codec [23]. The experiments are carried out at a Dell PC with CPU of Core 2 Duo 2.8 GHz, memory of 2 GB and OS of Windows XP.

In test 1, a 14-s-screen image sequence [24]

(resolution: 640×480 ; frame number: 214 frames; color depth: RGB 16 bit; data size: 125.4 MB) used by TechSmith to demonstrate its good compression performance is compressed with all testing codecs. Each codec is set to a frame rate of 15 frames per second to record screen data. PSNR is the mean of the PSNR for every frame since all frames of the sequence have approximate motion, texture and brightness, which give a reliable final result. CPU utilization is recorded every second for the duration of encoding period and then the data is calculated to produce average CPU utilization.

As seen in Table 1, the proposed CABHG method has approximately 48% higher compression ratio and 53% lower CPU utilization than Sorenson 3 codec. CAHG also has approximately 47% higher compression ratio and 46% lower CPU utilization than TechSmith Ensharpen codec.

Table 1 Evaluation of different codecs

Encoding tool	Encoded screen data/kB	PSNR/dB	CPU utilization/%
Sorenson 3 codec	301.03	35.7	37
TechSmith Ensharpen codec	292.60	∞	28
Apple's Animation codec	1 169.20	∞	42
MainConcept H.264 codec	1 288.00	33.4	79
XviD MPEG-4 codec	74.30	31.7	49
CABHG codec	156.30	41.6	15

CABHG also outperforms general video codecs such as Apple's Animation codec, H.264 codec and XviD MPEG-4 codec. CABHG method was approximately 87% higher compression ratio and 64% lower CPU utilization than Apple's Animation codec. CABHG has approximately 88% higher compression ratio and 81% lower CPU utilization than H.264 codec. CABHG has approximately 54% lower compression ratio and 69% lower CPU utilization than XviD MPEG-4 codec.

Both Apple's Animation codec and MainConcept H.264 codec cannot compress the video to a size suitable for real-time screen image sequence transmission. XviD MPEG-4 codec's lowest PSNR indicates its poor image quality after decoding, which is unacceptable in screen image sequence compressions. Apple's Animation codec and TechSmith EnSharpen codec with almost infinite PSNR are lossless codecs. Thus, their PSNRs are not considered in the comparison.

Due to unsuitability for the real time screen image sequence compression, four video codecs (Sorenson 3

codec, Apple's Animation codec, MainConcept H.264 codec, and XviD MPEG-4 codec) are skipped in further tests. Under almost the same satisfactory decoded quality, that is to say, no jagged edges and distorted texts, CABHG and Ensharpen are further compared in tests 2 and 3. In test 2, a 10-page PPT file is set to advance automatically every 50 s, typical operations in E-learning, and Ensharpen and CABHG's compression performances are evaluated, and the result is shown in Fig.8.

In test 3, a series of basic operations are performed, including compiling a document file with Microsoft Word and surfing the Internet. Performances of Ensharpen and CABHG are evaluated second by second, and the result is shown in Fig.9.

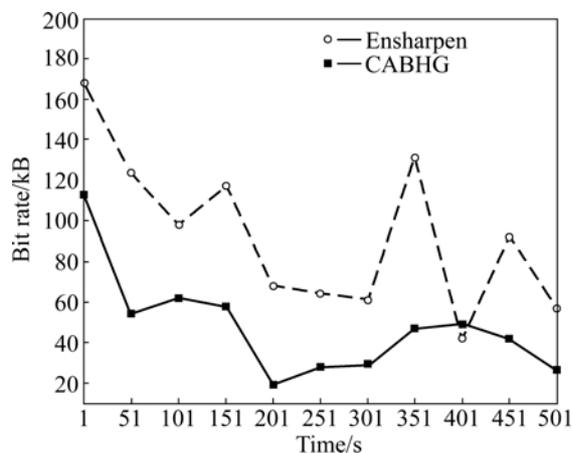


Fig.8 Evaluation between Ensharpen and CABHG in test 2

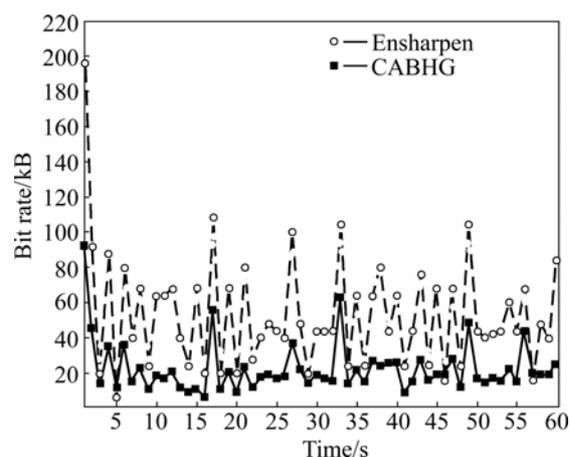


Fig.9 Evaluation between Ensharpen and CABHG in test 3

From tests 2 and 3, a conclusion can be safely drawn that the proposed CABHG produces less average screen data than Ensharpen codecs. Although CABHG is a lossy video codec, it makes a compromise between file size and video quality. The encoded data quality is not deteriorated and the proposed adaptive block size coding keeps edge information well, which is demonstrated by its PSNR. Furthermore, because of the use of mirror

video driver technology, CABHG reduces the image data required to be compressed, which leads to both low CPU utilization and small screen data.

6 Conclusions

(1) The proposed CABHG method is tailor-made for screen image sequences to reduce their temporal and spatial redundancy.

(2) An advanced DCT based variable block size intra-frame coding method and an improved hierarchical GOP structure are proposed to improve the system performance.

(3) Experimental results show that the CABHG outperforms not only the professional screen image sequence codecs but also general video codecs in both compression ratios and CPU utilizations.

References

- [1] TECHSMITH Corporation. EnSharpen video codec [EB/OL]. [2009-03-15]. <http://www.techsmith.com/download/codecs.asp>.
- [2] SORENSON Media Incorporated. Sorenson 3 codec [EB/OL]. [2009-02-24]. <http://www.sorensonmedia.com>.
- [3] ZHANG Shi-qiang, ZHANG Shu-fang, ZHAO Song, HU Qing. Image compression method based on image adaptive segmentation and adaptive quantification [J]. Journal of Dalian Maritime University, 2009, 35(1): 31–35. (in Chinese)
- [4] PONOMARENKO N N, EGIАЗARIAN K O, LUKIN V V, ASTOLA J T. High-quality DCT-based image compression using partition schemes [J]. IEEE Signal Processing Letters, 2007, 14(2): 105–108.
- [5] TAUBMAN D, MARCELLIN M. JPEG 2000: Image compression fundamentals, standards and practice [M]. Boston: Kluwer, 2002: 1–781.
- [6] WALLACE G K. The JPEG still picture compression standard [J]. Communications of the ACM, 1991, 34(4): 31–44.
- [7] LIU Dong, DING Wen-peng, HE Yu-wen, WU Feng. Quality-biased rate allocation for compound image coding with block classification [C]// Proceedings of the IEEE International Symposium on Circuits and Systems. Island of Kos: IEEE Computer Society, 2006: 4947–4950.
- [8] ZHOU Xian-cheng, SHEN Qun-tai, LIU Li-mei. New two-dimensional fuzzy C-means clustering algorithm for image segmentation [J]. Journal of Central South University of Technology, 2008, 15(6): 882–887.
- [9] DESHPANDE S G, HWANG J N. A real-time interactive virtual classroom multimedia distance learning system [J]. IEEE Transactions on Multimedia, 2001, 3(4): 432–444.
- [10] YIN Hao, LIN Chuang, ZHUANG Jin-jun, NI Qiang. An adaptive distance learning system based on media streaming [C]// Proceeding of the 3rd International Conference on Web-based Learning. Beijing: Springer, 2004: 184–192.
- [11] KOIVUSAARI J, TAKALA J, GABBOUJ M. Image coding using adaptive resizing in the block-DCT domain [C]// Proceedings of SPIE-IS&T Electronic Imaging Multimedia on Mobile Devices II. San Jose: SPIE, 2006: 607405–1–9.
- [12] WIEGAND T, SULLIVAN G J, BJONTEGAARD G, LUTHRA A. Overview of the H.264/AVC video coding standard [J]. IEEE

- Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 560–576.
- [13] DAI W, LIU L J, TRAN T D. Adaptive block-based image coding with pre-/post-filtering [C]// Data Compression Conference. Utah: IEEE Computer Society, 2005: 73–82.
- [14] HUANG Chun-ming, YANG Kai-chao, WANG Jia-shung. A low cost unrestricted fast playback scheme for video streaming [J]. IEEE Transactions on Circuits Syst. Part II: Express Brief, 2005, 52(7): 384–388.
- [15] MIN B, KIM S, MANDAL M, JEONG J. Efficient compression method for cell animation video [J]. IEICE Transactions on Communications, 2005, E88–B(8): 3443–3450.
- [16] LIM K W, CHUN K W, RA J B. Improvement on Image transform coding by reducing interblock correlation [J]. IEEE Transactions on Image Processing, 1995, 4(8): 1146–1150.
- [17] HSIA S C, YANG J F, LIU B D. Efficient postprocessor for blocky effect removal based on transform characteristics [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(6): 924–929.
- [18] WU C B, LIU B D, YANG J F. Adaptive postprocessors with DCT-based block classifications [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(5): 365–375.
- [19] DUGAD R, AHUJA N. A fast scheme for altering resolution in the compressed domain [C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Urbana: IEEE Computer Society, 1999: 213–218.
- [20] DUGAD R, AHUJA N. A fast scheme for image size change in the compressed domain [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(4): 461–474.
- [21] MAINCONCEPT GmbH. MainConcept H.264 codec [EB/OL]. [2009–3–17]. <http://www.mainconcept.com>.
- [22] XVID Organization. Xvid MPEG–4 codec [EB/OL]. [2009–03–07]. <http://www.xvid.org/>
- [23] APPLE Corp. Apple’s animation codec [EB/OL]. [2009–03–20]. <http://www.apple.com>.
- [24] TECHSMITH Corporation. Compression [EB/OL]. [2009–03–15]. <http://www.techsmith.com/codecs/ensharpen/compression.asp>.

(Edited by CHEN Wei-ping)