NOVEL REAL-TIME SOFTWARE-BASED VIDEO CODING ALGORITHMS

Ho-Chao Huang and Ja-Ling Wu
Communication & Multimedia Lab.,
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

Two novel real-time software-based video coders, the software-based moving picture coder (SBMPC) and the popular video coder (PVC), are proposed in this paper. SBMPC uses the modified block truncation code and the newly proposed multiresolution-in-time sampling techniques, PVC uses the adaptive quantizer and the modified windowed Huffman-kind coder, to make both the coding speed and the compression ratio much higher than those of traditional video coders. By using the proposed coders, video quality and coding speed are good and fast enough for practical video applications. Since no compression hardwares are needed in the proposed video coders, the cost and complexity of developing multimedia applications can be greatly reduced.

1 Introduction

In multimedia applications, to efficiently compress the video data is one of the most important functions. Some existing video coding techniques, such as H.261[1], RM8[2] and MPEG[3], have been designed to compress video data by both reducing the spatial correlation through the transform coding, such as DCT[4], and the temporal correlation through the DPCM[5] and motion compensation (MC)[6] techniques. One of the major problems of those techniques is the imposed high computational complexity, which inhibits those techniques for real-time applications especially by software.

In order to release the dependency of the video applications to expensive compression hardwares, two software-based video coders, the software-based moving picture coder (SBMPC) and the popular video coder (PVC), are proposed in this paper. In SBMPC, two new techniques, the modified block truncation code and the multiresolution-in-time (MIT) sampling, are used. The block truncation coding[7] (BTC) is considered as an edge-preserving algorithm for image coding and is a relatively fast algorithm as compared with the transform coding. In order to speed up SBMPC further, a modified version of BTC is proposed. In order to increase the compression ratio, the MIT sampling technique is adopted. The MIT sampling technique analyzes the temporal activity of the image sequence and assigns a higher sampling rate to the blocks which possesses higher degree of activity, and vice versa. By using these new techniques, about 20 to 25 compression ratio and about 14 frame/second compression speed can be obtained on an IBM PC for a color video signal with $128 \times 128$ frame size.

PVC combines the DPCM technique, the adaptive quantizer and the modified windowed Huffman-kind algorithm (which is a modified version of the windowed Huffman algorithm proposed in [8]) to compress the video data. By using these techniques, PVC can compress the video data swiftly and efficiently. The range of the compression ratio of PVC is from several to hundreds. In the same picture resolution as used in SBMPC, the average compression ratio, peak SNR and the average frame rate are about 34, 30 dB, and 20 frame/second respectively. The picture quality of PVC is even better than that of MPEG coder, under the same picture resolution and therefore is good enough for commercial uses and is good especially for head-and-shoulder video applications. The coding speeds of both SBMPC and PVC are much faster than those of MPEG and are also fast enough to meet the real-time constraint of commercial video applications. Moreover, no block, ripple and blurring effects are produced by PVC.

This paper is organized as follows. Section 2 reviews the existing MC/DCT-based video coder, the block truncation code and the windowed Huffman coder, briefly. SBMPC and PVC are presented in sections 3 and 4, respectively. Section 5 shows the experimental results. Conclusions, discussions and future works are addressed in section 6.

2 Preliminary Reviews
2. MC/DCT-Based Video Coder

The basic idea of MC/DCT-based video coders is to reduce the bit rate by using the source coding to remove the spatial and temporal redundancies, and using the entropy coding to remove the codeword redundancies. A typical source coder usually consists of a predictor and a transform coder. The prediction coder is used to remove the temporal redundancies, and the transform coder is used to remove the spatial ones. A typical entropy coder usually contains a run-length coder and a Huffman coder. The run-length coder encodes the zeros in succession of the quantized transform coefficients, and the Huffman coder represents the run-length coded tuples in the most compact form. The block diagram of H.261 encoder is shown in Fig. 1.

Although MC/DCT-based video coders have the capability of encoding moving pictures with relatively high compression ratio and fair picture quality, the following problems of this coding scheme remain.

1. High computational complexity is required.

   This is the major problem of MC/DCT-based video coders. Enormous operations are needed in order to perform the functions of the motion estimation and the DCT. Even though fast algorithms of MC and DCT are used, the processing is still too slow to be applied for real-time multimedia applications, by using software based MC/DCT video coders. Meanwhile, the hardware solution of the video coder also has the following disadvantages.

   (a) The cost of the compression chips is very expensive.

   (b) The development of the device drivers is time consuming and not trivial.

   (c) It is inconvenient to install, maintain, and upgrade video applications by using sophisticated compression chips.

2. Block effect exists.

2.2 Block Truncation Code

In BTC, an image is first divided into \( n \times n \) nonoverlapping blocks, and then these blocks are quantized by a two-level block-dependent quantizer. Both the quantizer's threshold and the two reconstruction levels are adaptive in response to the local statistics within a block. An example of BTC is given in figure 2.

The advantages of BTC are listed below, and these are the reasons BTC was chosen as the basic kernel of SBMPC.

1. The coding speed of BTC is much faster than that of transform coding.

2. The BTC is an edge-preserving coding algorithm.

Although the BTC has the benefits described above, there are some inborn defects existed in BTC and these are the reasons that why BTC can not be used directly in SBMPC:

1. It is not easy to adapt the compression ratio and the quality of BTC when the block size is fixed.

2. The compression ratio of BTC is not high enough to meet the bit rate requirement of a video coding system.

3. Although the coding speed of BTC is faster than that of DCT, it is still not fast enough to meet the real-time requirement of SBMPC system.
The modified BTC and MIT sampling, which are used in SBMPC system and will be introduced in section 3, are designed to overcome the limitations of BTC mentioned above.

2.3 Windowed Huffman Coder

The original idea of this algorithm was proposed in [9][10] and a similar coding scheme applied to arithmetic coding was proposed in [11]. The code tree of the windowed Huffman algorithm is constructed by adjusting the tree when the weight of a node is increased or decreased. The weight of a leave node in the windowed Huffman algorithm is the number of occurrences of the associated symbol in the limited past history. In windowed Huffman algorithm, a window with size \( n \) is used to store the most recently encoded \( n \) symbols. When encoding a symbol, the symbol is added into the window, then the weight of the corresponding node is increased, and the Update-Increase procedure, which is similar to the Update procedures of algorithm V[10] and algorithm FGK[12], is performed to adjust the windowed Huffman tree. When the window is full, the oldest symbol in the window is removed, the weight of the node is decreased, and the Update-Decrease procedure, which is similar to the Update procedure of residual Huffman algorithm[13], is performed to readjust the code tree. If the size of the window is selected properly, the coding efficiency of the windowed Huffman algorithm is the best among all static and dynamic Huffman algorithms.

Although the performance of the windowed Huffman algorithm is the best, the processing speed of the algorithm is not fast enough to meet the real-time constraint of video applications. It follows that some modifications to speed up the coding process is needed and the modified windowed Huffman-kind algorithm, detailed in section 4.5, is so obtained.

3 Software-Based Moving Picture Coder (SBMPC)

The goals of SBMPC system are to solve the time complexity problem of MC/DCT-based coder by proposing a new high-speed coding architecture instead of hardware chips solutions, and to obtain a high-quality compressed video with high compression ratio and edge-preserving (object-preserving) characteristics.

3.1 Modified BTC

In modified BTC, the input image is first segmented into \( 8 \times 8 \) blocks and then the pixels within a block are subsampled. This subsampling process is necessary because it can both accelerate the computations and achieve higher compression ratio. Furthermore, a simplified algorithm is used for computing the quantizer’s threshold and the two reconstruction levels for each subsampled block, and then the pixels are quantized by a two-level quantizer. The decoder of modified BTC reconstructs the quantized pixels, interpolates the pixels eliminated by the subsample process, synthesizes and generates the output image. The block diagram of modified BTC is shown in figure 3.

The Blocker of the modified BTC is used for dividing the original image data \( OX(n) \) into blocks \( X(n) \) with size \( 8 \times 8 \). These blocks are then sent into the Subsampler. In SBMPC, \( k \) is set to 3 so as to get the information of each horizontal and vertical scan lines. Figure 4 shows an example of the function performed by Subsampler.

The QP Decider are used to decide the quantizer’s threshold \( QT \) and the two reconstruction levels \( RL(0) \) and \( RL(1) \) of the two-level quantizer. In order to accelerate the computation speed of QP Decider, the \( SX(n) \), which is generated by Subsampler, is used instead of the original image blocks \( X(n) \) as the input of QP Decider. The parameters, \( QT \), \( RL(0) \) and \( RL(1) \) are defined as:

\[
QT = \frac{1}{\|X\|} \sum_{\|i\| < \|X\|} SX(i)
\]

\[
RL(0) = \frac{1}{\sum_{\|i\| < \|X\|} SX(i)} \sum_{\|i\| < \|X\|} \delta_{\|i\| < QT} SX(i)
\]

3.1 Modified BTC

In modified BTC, the input image is first segmented into \( 8 \times 8 \) blocks and then the pixels within a block are subsampled. This subsampling process is necessary because
Huang and Wu: Novel Real-Time Software-Based Video Coding Algorithms

Figure 4: An example of execution process of Subsampler with \( k = 3 \), (a) is the input image data, and the framed pixels will be selected by the Subsampler, and (b) is the output data.

\[
\begin{align*}
&\text{RL}(1) = \frac{1}{\sum_{\psi} \delta_{\psi} \sum_{0 \leq (\psi \text{mod } 8) < QT} \delta_{X(i)}}, \quad (4) \\
\text{where } k &\text{ is the same as the one defined in Subsampler, and } \delta_f = 1 \text{ when } f \text{ is true, otherwise, } \delta_f = 0. \\
&\text{The Decimator performs vertical and horizontal decimation functions, which are similar to the subsample function of Subsampler. These functions are defined as follows:} \\
&1. \text{horizontal decimation function:} \\
&DX(n) = DX_h(n) \\
&\quad = X(d \times n), \quad d \text{ is positive} \quad (5) \\
&2. \text{vertical decimation function:} \\
&DX(n) = DX_v(n) \\
&\quad = X(\left\lfloor \frac{n}{w} \right\rfloor \times d \times w + (n \text{ mod } w)) \quad (6)
\end{align*}
\]

where \( w \) is the width of the image block, \( \lfloor r \rfloor \) returns the closest integer to \( r \) of the integers which are less than or equal to \( r \).

The Two-Level Quantizer quantizes the decimated image \( DX(n) \) output by Decimator. The quantizer’s threshold \( QT \) defined by equation (2) is used for the quantization process. The bit-map \( B(n) \) output by the Two-Level Quantizer is defined by:

\[
B(n) = \delta_{DX(n)\geq QT} \quad (7)
\]

The bit-map \( B(n) \) is then multiplexed with the reconstruction levels \( RL(0) \) and \( RL(1) \) and sent to the modified BTC decoder.

In the modified BTC decoder, the Inverse Quantizer performs an inverse function of Two-Level Quantizer.

Because the image pixels reconstructed by the Inverse Quantizer are the decimated pixels, the pixels eliminated by the Decimator must be recovered by the Interpolator. Finally, the recovered image blocks \( X(n) \) are synthesized into an output image \( \hat{X}(n) \) by the Image Synthesizer.

### 3.2 MIT Sampling

Although the compression ratio of modified BTC is higher than traditional BTC algorithms, it is not high enough to meet the bandwidth requirement of video coding. In order to obtain a higher compression ratio than the modified BTC, the MIT sampling technique is adopted. There are several observations which motivate the development of the MIT sampling technique.

**Observation 1** The adjacent images are similar to each other. The DPCM technique can remove most of the time redundancies in the image sequence.

The observation 1 shows a trivial fact that most of the time redundancy can be removed by DPCM technique. When the image sequence is slow variant, the residual image obtained from the output of DPCM contains little information and can be discarded. When the image sequence contains a scene change or some objects in the image sequence move quickly, the residual image of the DPCM would contain a lot of information and must be coded by the modified BTC. The following observations can assist us to code these residual images more efficiently.

**Observation 2** The residual images of DPCM are segmented into blocks in the SBMPC system. The activities of these residual image blocks are space-variant.

The activity \( ACT \) of an image block is defined as:

\[
ACT = RL(1) - RL(0), \quad (8)
\]

where the \( RL(0) \) and \( RL(1) \) are defined in equations (3) and (4) respectively.

**Observation 3** The quickly variant residual blocks, such as which contains the outlines of moving objects, usually have higher \( ACT \)’s than that of the slowly variant residual blocks. Therefore, the information contained on the blocks with higher \( ACT \)’s are considered more important than that of the blocks with lower \( ACT \)’s.

When the objects in the video are moving, it is more important to trace the actions of the objects fast and clearly than to describe the contents of these objects in detail. Because the residual image blocks which contain a part of moving objects usually have higher \( ACT \) values than the blocks which contains a part of background...
image, the MIT sampler codes the higher activity blocks with higher sampling rate in the time axis and vice versa.

In SBMPC system, the MIT sampler is designed as follows. The residual image blocks are classified into several activity classes according to the activities of the blocks. If all the blocks within the same activity class are sampled at the same time instance, it will make the video sequence discontinuous. For this reason, instead of sampling these image blocks in time axis, the SBMPC system subsample the residual image blocks by block sequence. For example, if the sampling rate of some activity class is 3 blocks/sample then only the 1st, 4th, 7th, ..., blocks in the activity class are coded. In order to sample these image blocks fairly, the starting sampled block is randomly selected. That is, the MIT sampler samples the image block sequences '1, 4, 7, ...', '2, 5, 8, ...', and '3, 6, 9, ...' with equal probability.

Combining the modified BTC and the MIT sampling techniques, SBMPC system can code the moving pictures with high coding speed and high compression ratio.

### 3.3 SBMPC System

The block diagram of SBMPC encoder is shown in figure 5. The input image is first partitioned into image blocks by Blocker. These image blocks are then sent into PCM and DPCM, simultaneously. For each input image block, two image blocks, one is the image block output from PCM and the other is the residual image block generated by the DPCM, are obtained. $RL(0)$s and $RL(1)$s of two image blocks are calculated, called $RL_P(0)$, $RL_P(1)$ and $RL_D(0)$, $RL_D(1)$, respectively. Furthermore, PCM/DPCM Multiplexer decides which image block must be sent to MIT Sampler based on the following function:

$$ Block = \begin{cases} Block_{PCM}, & \text{if } (RL_P(1) - RL_P(0)) \\ Block_{DPCM}, & \text{otherwise} \end{cases} $$

where $Block$ is the output image block of PCM/DPCM Multiplexer, $Block_{PCM}$ means the output image block of PCM, $Block_{DPCM}$ is the output image block of DPCM, and $TH_{PID}$ is the threshold used for PCM/DPCM selection.

The MIT Sampler gets the image block sent by the PCM/DPCM Multiplexer and samples it or not based on the MIT sampling rules. If the input image block is selected by MIT Sampler, modified BTC algorithm is applied to encode this image block. Otherwise, the mean value of this image block is calculated. The output data of the modified BTC and the mean values are then sent to SBMPC decoder. The SBMPC decoder reconstructs the compressed data, and its block diagram is shown in figure 6.

### 4 Popular Video Coder (PVC)

The block diagram of PVC encoder is shown in Fig. 7. In PVC, the input video sequence is organized as groups of frames (GOFs). There are four frame types within a GOF, which will be described in subsection 4.2, and they are controlled by the Frame Type Controller. The type of frame affects the functions of Adaptive Quantizer, Frame Buffer and Modified Windowed Huffman-Kind Coder.

In order to meet the real-time constraint of video applications, PVC discards the DCT coder and the motion compensator which are widely used in MC/DCT-based...
Huang and Wu: Novel Real-Time Software-Based Video Coding Algorithms

increase the coding efficiency further, a fast version of Y, U and V components of the image are encoded independently. The Y, U and V components of the image are subtracted from the reconstructed previous frame stored in the Frame Buffer, quantized by the Adaptive Quantizer, and then encoded by the Run-Length Encoder and the Modified Windowed Huffman-Kind Coder. Meanwhile, the quantized Y, U and V components are dequantized (reconstructed) and stored in Frame Buffer, to be used as the reconstructed previous frame in encoding the next frame. The decoding process of PVC is similar to the reverse process of the encoder and its block diagram is shown in Fig. 8. The following subsections will depict the function blocks of PVC in details.

4.1 Format Convertor

The Format Convertor converts the RGB components of pixels to YUV components. After the format conversion, the U and V components are then 4 to 1 subsampled. The functions of format conversion are:

\[
\begin{align*}
Y(x) &= 0.299 \times R(x) + 0.587 \times G(x) + 0.114 \times B(x) \quad (10) \\
U(x) &= (B(x) - Y(x)) \times 0.493/1.74 \quad (11) \\
V(x) &= (R(x) - Y(x)) \times 0.877/2.46 \quad (12)
\end{align*}
\]

where \(R(x), G(x), B(x), Y(x), U(x),\) and \(V(x)\) are the values of red, green, blue, Y, U, and V components of pixel \(x\), respectively.

4.2 Frame Type Controller

The Frame Type Controller defines the organization of GOF, sets the type of each frame, and controls the parameters of Adaptive Quantizer, Frame Buffer and Modified Windowed Huffman-Kind Coder. There are four types of frames within a GOF: Refresh Frame (R-Frame), Fine Frame (F-Frame), Medium Frame (M-Frame) and Coarse Frame (C-Frame). The use of R-Frame is just for the error control of the communicated video, and it is only used in the beginning of a GOF. Excluding R-Frame, F-Frame has the smallest compression ratio but possesses the best picture quality. The most gain of the compression ratio comes from C-Frame, but the picture quality of C-Frame is not so good as that of F-Frame. The picture quality and compression ratio of M-Frame are in between those of F-Frame and C-Frame. The use of the M-Frame is the tradeoff between the high picture quality and the high compression ratio. Following are their descriptions.

Refresh Frame:

In R-Frame, the Frame Buffer is cleared, the Huffman table of Modified Window Huffman-Kind Coder is reset, and the quantization steps of the Adaptive Quantizer are set as similar to those in F-Frame. The type of the first frame in a GOF is always set as the R-Frame.

Fine Frame:

In F-Frame, the quantization steps are the smallest of all four frame types. Therefore, the picture quality of the F-Frame is the best of all.

Medium Frame:

In M-Frame, the quantization steps are larger than those in F-Frame.

Coarse Frame:

The quantization steps used in the C-Frame are the largest, and the picture quality of C-Frame is the worst.

The parameters to control the Frame Type Controller can be written as the vector \(FT\): \((GL, FL, ML)\), where \(GL\) indicates the length of GOF, \(FL\) sets the frame distance of two F-Better-Frames (where an F-Better-Frame is defined as an F-Frame or an R-Frame), and \(ML\) sets the frame distance of two M-Better-Frames (where an M-Better-Frame is defined as an M-Frame or an F-Frame or an R-Frame). The number of C-Frames between any two other type frames is equal to \(ML - 1\). A typical organization of GOF is shown in Fig. 9, and the vector \(FT\) is chosen to be \((12,8,3)\).
4.4 Run-Length Encoder

The Run-Length Encoder represents the quantized values as the run-length tuples \( RL(Z, V) \), where \( Z \) indicates the length of zeros in succession, and this zero run is followed by a quantized value \( V \).

4.5 Modified Windowed Huffman-Kind Coder

After the quantized signal is encoded as the run-length tuples, these tuples are then represented in more compact form by using Huffman code[14]. Usually, the probability distribution of each symbol is time variant; hence, the concept of the windowed Huffman algorithm[8] is applied. In order to speed up the coding process to meet the real-time constraint, Modified Windowed Huffman-Kind Coder modifies the Initial and the Update procedures of [8], which are used to construct and adjust the Huffman codetree, respectively.

The maximum run-length used in Run-Length Encoder is 15 such that 4 bits are needed for representing each run-length. The range of a quantized value in Adaptive Quantizer is from -7 to 8 such that 4 bits are needed to represent it, too. In PVC, the two fields of a run-length tuple \( RL(Z, V) \) are combined as an 8-bit symbol. That is, the alphabet size of Modified Windowed Huffman-Kind Encoder is 256. In order to simply the encoding procedure, the concept of the length-limited Huffman code[15][16] is used. In order to speed up the process of code tree reconstruction, a simplified version of the Concept of the length-limited Huffman code[15][16] is used. In order to speed up the process of code tree reconstruction, a simplified version of the Construction procedure is adopted in PVC. The maximum code length of encoded symbols is limited to 12 bits.

The encoding process of Modified Windowed Huffman-Kind Encoder is briefly described as follows. Before encoding each symbol, the encoder first checks whether it is the time to reconstruct the Huffman codetree. If time is up, the encoder calls the Construct-CodeTree procedure to reconstruct the Huffman codetree. In PVC, the Huffman codetree is reconstructed for every three frames. After the reconstruction process, the next input symbol is encoded by the Encode procedure and the code table is adjusted by Update procedure. The Update procedures proposed in [8] and [13] can be used to adjust the Huffman codetree, but in view of computation speed, the Update procedure in Modified Windowed Huffman-Kind Encoder removes the part of codetree adjustment, and increases the weight of encoded symbol only.

5 Experimental Results

This section shows several experiment results. In these experiments, the resolution of each image is \( 128 \times 128 \).
Table 1: The DPCM utilities, the average compression ratios, and the average peak SNRs of SBMPC for encoding the salesman sequence, when the $TH_{PID}$ value varying.

<table>
<thead>
<tr>
<th>$TH_{PID}$</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPCM util. (%)</td>
<td>25.2</td>
<td>27.7</td>
<td>72.4</td>
<td>97.3</td>
<td>98.8</td>
<td>99.3</td>
<td>99.5</td>
</tr>
<tr>
<td>Comp. ratio</td>
<td>12.0</td>
<td>12.0</td>
<td>13.2</td>
<td>12.5</td>
<td>12.3</td>
<td>12.1</td>
<td>12.0</td>
</tr>
<tr>
<td>Peak SNR</td>
<td>29.0</td>
<td>29.5</td>
<td>29.7</td>
<td>29.3</td>
<td>29.3</td>
<td>29.2</td>
<td>29.3</td>
</tr>
</tbody>
</table>

In order to evaluate the coding speeds of SBMPC and PVC, two time intervals are measured. One is the gross coding time including the frame grab time, encoding time, transmission (saving) time, and the time interval for playing back to the display. The other is the net coding time including the encoding time and the transmission time only. In SBMPC, the average gross coding time is 141 ms/frame and the average gross coding speed is about 7 frame/sec. In PVC, the average net coding time is 34.617 ms/frame which results in an average net coding speed of about 29 frame/sec. The average gross coding speed at about 17 frame/sec.

The first experiment shows the effects caused by adjusting the parameter $TH_{PID}$ of SBMPC. In this experiment, the $TH_{PID}$ varies from -3 to 3. Table 1 shows the DPCM utilities, the average compression ratios, and the average peak SNRs with respect to these distinct $TH_{PID}$ values.

The second set of experiments is to evaluate the PSNRs and average compression ratios of the PVC by changing the value of $FL$, where the values of RL and ML respectively are set to be 256 and $FL/3$. In these experiments, two average compression ratios are evaluated. One is the average compression ratio (ACR) of the entire video sequence, and the other is the average communication compression ratio (ACCR) of the video sequence except the refresh frame. In communication applications, the refresh frames are only used to initialize PVC and to recover from transmission errors. They only affects the setup time of PVC, and will not add any traffic load to the network in normal status. Therefore, for communication applications, to evaluate ACCR is more meaningful than to evaluate ACR of PVC. Fig. 11 shows the PSNRs, ACRs and ACCRs of encoding the salesman and Chinese girl sequences. These figures show that the larger the $FL$ is, the lower the average PSNR and the higher the ACCR would be.

The final experiment is to compare peak SNRs (PSNRs) to those of SBMPC, PVC and MPEG coders for encoding the salesman sequence. As the experimental results shown in figure 12, the PSNRs of PVC are even higher than those of MPEG coder. The original video sequence and the sequences encoded by SBMPC, PVC and MPEG coders are shown in figure 13.

6 Conclusions, Discussions and Future Works

This paper presents two novel software-based video coders, the software-based moving picture coder...
Figure 13: The comparisons of the SBMPC, PVC and MPEG coder for the 29th, 30th, 32nd and 33rd frames of salesman sequence. The compression ratio for those coders is 34.
Huang and Wu: Novel Real-Time Software-Based Video Coding Algorithms

Figure 12: The PSNRs to compression ratios of SBMPC, PVC and MPEG.

(SBMPC) and the popular video coder (PVC). SBMPC uses the modified BTC and MIT sampling techniques; while, PVC simplifies the MC/DCT-based video coders to meet the real-time constraint of video applications. PVC also modifies the quantizer and the Huffman coder to achieve high compression ratio with good picture quality.

As the results shown in previous section, the coding speed and the picture quality of PVC are the best of all. The advantages of the SBMPC are that it is easy to be implemented, and the quality of the scene changed pictures is better than that of PVC. The picture quality and the coding speed of the proposed coders are suitable for the commercial multimedia applications. By using SBMPC or PVC as the video coder, both the cost and the complexity of developing, maintaining, and installing of video applications can be greatly reduced.

When the computation power of PC is increased in the future, the performances of PVC and SBMPC can be improved in several ways. First, the windowed Huffman algorithm[8] can be completely applied to encode the data more compactly. Second, some simple and fast motion estimation algorithm can be used to remove the temporal redundancies further. Third, some fast transform algorithms can be used to remove the spatial redundancies. Fourth, entropy coding can be introduced into SBMPC system to encode data more compactly. Finally, the frame rate can be increased and the frame size can be enlarged.

Acknowledgment

This work is supported by National Science Council, Taiwan, R.O.C., under grants no. NSC82-0408-E-002-232.

References


Biographies


Ho-Chao Huang was born in Ilan, Taiwan, R. O. C., on December 19, 1968. He received the B.S. degree in the Department of Computer Science and Information Engineering from the National Taiwan University in 1990. He is now a Ph.D. student in the Department of Computer Science and Information Engineering at the National Taiwan University, Taipei, Taiwan.

His research interests include Variable Length Codes, Software-Based Video Data Compression, Model-Based Image Coding, Computer Networks, Computer Graphics, Graphics User Interface and Multimedia Systems. He is a student member of IEEE.

Ja-Ling Wu was born in Taipei, Taiwan, on November 24, 1956. He received the B.S. degree in Electronics Engineering from the Tamkang University, Tamshoei, Taiwan, in 1979, the M.S. and Ph.D. degree in Electrical Engineering from Tatung Institute of Technology in 1981 and 1986, respectively.

From 1986 to 1987 he was an Associate professor of the Electrical Engineering Department at Tatung Institute of Technology, Taipei, Taiwan. Since 1987 he has been with the Department of Computer Science and Information Engineering, National Taiwan University, where he is presently a Professor.

Dr. Wu was the recipient of the 1989 Outstanding Youth Medal of the Republic of China and the Outstanding Research Award sponsored by the National Science Council, from 1987 to 1992.

Dr. Wu has published more than 100 technique and conference papers. His research interests include Neural Networks, VLSI Signal Processing, Parallel Processing, Image Coding, Algorithm Design for DSP, Data Compression, and Multimedia Systems.