Algorithmic Implementation and Efficiency Maintenance of Real-Time Environment using Low-Bitrate Wireless Communication

A. Jalal and S. Kim
School of Electrical Engineering and Computer Science
Kyungpook National University, Korea
jalal@woorisol.knu.ac.kr, kimsw@knu.ac.kr

Abstract

This paper presents different issues of the real-time compression algorithms without compromising the video quality in the distributed environment. The theme of this research is to manage the critical processing stages (speed, information lost, redundancy, distortion) having better encoded ratio, without the fluctuation of quantization scale by using IP configuration. In this paper, different techniques such as distortion measure with searching method cover the block phenomenon with motion estimation process while passing technique and floating measurement is configured by discrete cosine transform (DCT) to reduce computational complexity which is implemented in this video codec. While delay of bits in encoded buffer side especially in real-time state is being controlled to produce the video with high quality and maintenance a low buffering delay. Our results show the performance accuracy gain with better achievement in all the above processes in an encouraging mode.

1. Introduction

Real-time Video Codec (RVC) with compression efficiency and reduction in computational complexity with the improved video quality is the need of user's today. Therefore, a progressive enhancement in performance (video size, PSNR, speed) of video codec with the reduction of complexity is one of the major issues of research in this paper. In the conventional researches, the focusing point of RVC is to make compression ratio more efficient by considering in particular areas (accuracy, rate-distortion). In this paper, however we propose the viewer's attention towards the block behavior (distortion, matching, searching) which effects the processing speed, the control the floating point which makes relation with accuracy, the video quality and the compression efficiency all move in parallel direction.

While sensitive areas (recording noise, information deduction) are properly operated by considering the distortion measure which arranges the impressive quality view, then the encoder buffer receives the compressed bitstreams and controls their flow by using the advanced coding control [8] to maintain low buffer delay which causes reduction in skipping frame rate with improved PSNR quality. Some papers[9,10] were designed for the blocking effects are insufficient to perform well with critical information lost which causes visible artifacts at the real-time decoded video.

In this paper, the main goal is to achieve the best quality of video with improved form of compression. As to consider H.263 (tmn8) codec [6], a similar work [1] related with coding achievement, has enhanced their performance (PSNR) upto 20-23% by considering some extra reward of the distortion measure mechanism with full fast searching algorithm. Our initial performance results related with encoded (compressed size and video quality) process are compared with the popular H.263 codec by Telenor [3] and H.263 Conversational High Compression (CHC) encoder and show some uniform progressing achievement with same environment. Simultaneously, the fluctuation and fast implementation is ideal for A/V compressed delivery system and for video conferencing applications.

The paper is organized as follows. In the second section we describe the features of video compression by concentrating on their accuracy and processing factors. In the third and fourth sections, we explain the video decoding process and the overall performance with some comparison with other codec. Finally, in the fifth section, we conclude with future directions.

2. Real-Time Video Encoder of H.263

In this section, we describe the real-time and efficient video encoder of H.263.

Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)
0-7695-2560-1/06 $20.00 © 2006 IEEE

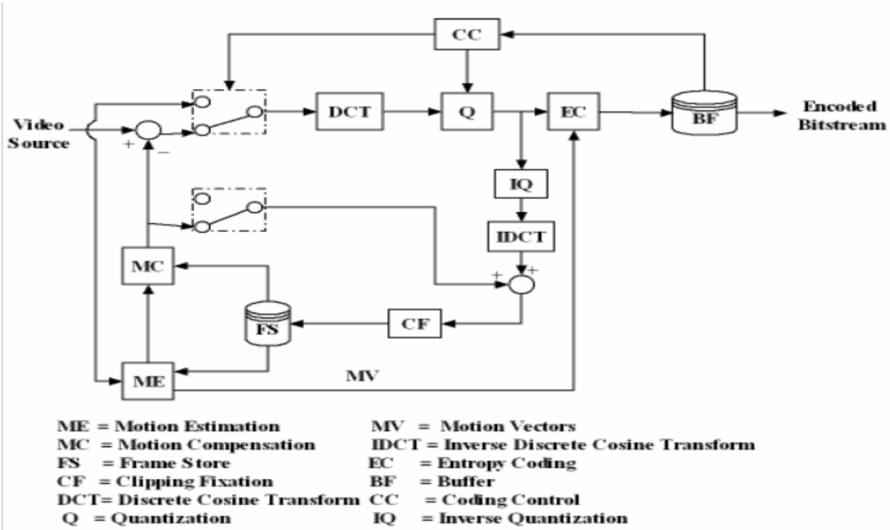

Figure 1. Data Flow Structure of the H.263 Video encoder

A. Structure of Real-Time Video Encoder

The H.263 video encoder is used to transfer video file in compact form by considering the features like removal of spatial and temporal redundancies within the frames, concentration on improving the quality of the video and improving rate distortion performance. H.263 encoder is described by using the block diagram in Figure 1. The coding control makes a relationship with the buffer by controlling (up and down flow) output bitstream in the real-time video transmission. Motion estimation and compensation, discrete cosine transform (DCT), quantization and entropy coding are the four main pillars on which the encoder performs most of its execution time.

B. Motion Estimation and Compensation

Motion mechanism in the video coding achieves more competent effects by reducing the temporal redundancy. In the motion estimation, the MxN blocks of the current frame are compared with that of the previous frame to find the "best matched" block between both the frames. While in motion compensation, residual is produced and transmitted by the help of motion vector. Motion estimation is based on the block matching algorithm (BMA) which easily overcomes all difficulties such as minor elimination changes in frames, weaker judgments along moving edges and motion vector unreliability. Selecting the best block can also faces another issue in the form of block distortion which deflates a lot of execution time to process and noise effects are briefly explained.

Considering the block behavior activities, we focus our operations on the factors like edge detection process, block distortion measure with distance criteria and searching method [2] which show its performance accuracy used in further processes.

Overall, operations performed in this section are explained by considering Figure 2.

In this paper, the block distortion measure (BDM) is used such as difference of the chrominance or texture, intensity change and the edge error. Boundaries of the object always contain critical edges. To make improvement and accuracy in the process of selecting best block, we use some edge accuracy formulation. Edge information is not limited for the matching accuracy but it conveys information (texture, object) from the block (previous block) to the current block.

$$E(d) = \sum_{i,j} w_{i,j} \, |(E_{(i,j)}, t) - (E_{(x+i,y+j)}, t+1)|^2 \qquad (1)$$

E (d) is the edge difference for the moving window which shifts by a small distance in the points*(x, y)*. Similarly, the average change of intensity and

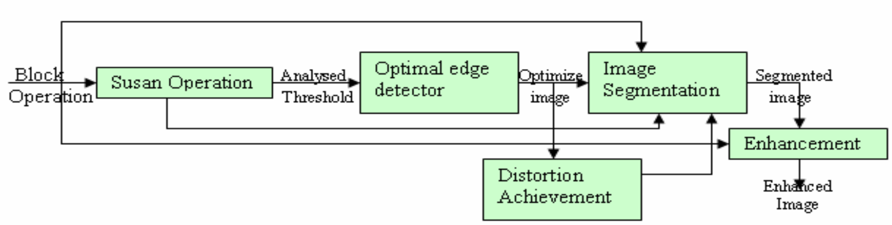

Figure 2. Outline of proposed Enhancement algorithm

Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)
0-7695-2560-1/06 $20.00 © 2006 IEEE

chrominance are considered as:

$$N_w(d) = \sum_{i,j} w_{i,j} \mid (I_{(i,j)}, t) - (I_{(x+i,y+j)}, t+1) \mid^2 \quad (2)$$

I is defined as the input image in the frame at difference position along with time t.

$$C(d) = \sum_{i,j} w_{i,j} [\mid (U_{(i,j)}, t) - (U_{(x+i,y+j)}, t+1) \mid^2 \quad +$$

$$\mid (V_{(i,j)}, t) - (V_{(x+i,y+j)}, t+1) \mid^2] \quad (3)$$

So the matching between the two blocks is represented as follows:

$$M(d) = (1/\text{MxN}) * (Nw(d) + E(d) + C(d)) \quad (4)$$

The procedure to find the best matched block is surrounding around the full and fast searching algorithms but their characteristics (time consumption, accuracy, quality) are injected as favorable for one and opposite to another. So due to this fact, fast algorithm with spiral search is implemented [4]. In spiral search motion, the search starts from the center and moves in spiral order. To predict the search window center, spiral search uses the motion vectors of the predictor blocks. The block having best matching selection using motion vector should be around this predicted center, if the spatial correlation operation is performing successfully. Using the motion vector, sum of absolute difference (SAD) of operating macroblock is calculated line by line. The comparison of the current and the previous SAD of every line are used to give the minimum SAD (same flow of macroblock) and to reject the macroblock if the current macroblock has greater SAD than the previous one.

C. Discrete Cosine Transform

Discrete Cosine transform (DCT) plays a vital role in the compression and decompression standards which processes block of pixels into frequency domain coefficients. It shows efficient performance in terms of energy compression potentiality by using fast algorithms (Feig & Winogard, LLM and AAN) that reduce the computational cost. 1D-DCT having 8 elements require 5 multiplication and 29 addition operations developed by Arai-Agui-Nakajima (AAN) [5].

While operation carried out on the 2D-DCT containing 464 additions and 144 multiplications for 8x8 elements.

Forward 8x8 2D DCT has the equation as follows:

$$F(u,v) = \frac{Cu}{2} \frac{Cv}{2} \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) Cos \frac{\pi u(2i+1)}{16} Cos \frac{\pi v(2j+1)}{16}$$

f i, j and the corresponding inverse 8x8 2D DCT, defined as follow:

$$f(i,j) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} Cos \frac{\pi u(2i+1)}{16} Cos \frac{\pi v(2j+1)}{16} CuCvF(uv)$$

Addition and subtraction operation has been performed well with the integer input, but DCT has encountered computational complexities when multiplication has been done with the fractional constants. To solve these problems following steps have to be performed.

(a) Multiply all the constants (sin [i, j], cos [i, j]) by applying value fixation operation.
(b) Scale a fraction constant in DCT-overscale process. Such a constant can be multiplied with an overscaled input to produce something that scaled by DCT-scalar.
(c) For proper rounding, divide the result (after multiplication) to obtain DCT scale which finally apply lifting shift process [5] to execute into integer value.

Shifting and adding functions implementation attain fast integer for the transform coefficient.

D. Quantization

After DCT, quantization has performed on the AC and DC coefficients to reduce the number of bits and to increase the number of zero-value coefficients which has reduced recording noise while improving bandwidth performance.

Equation for the AC coefficients of the intra block used in the quantization is

LEVEL = abs (coeff) / 2 * QP

While the DC coefficient of the intra block is quantized by using the step size of eight. It is established from the 8x8 matrices.

Equation for all the coefficients of inter blocks is

LEVEL = (abs (coeff) – QP/2) / 2 * QP

LEVEL is the quantized transform coefficient. Coeff is the transform coefficient that is to be quantized. QP is the quantization parameter which ranges the integer values from 1 to 31 which manipulate the scale factor code.

Inverse quantization is processed to make the reconstructed coefficient. If LEVEL = “0” then reconstructed level REC = “0”. The reconstructed level [7] of all non-zero coefficients other than the

Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA’06)
0-7695-2560-1/06 $20.00 © 2006 IEEE

IEEE COMPUTER SOCIETY

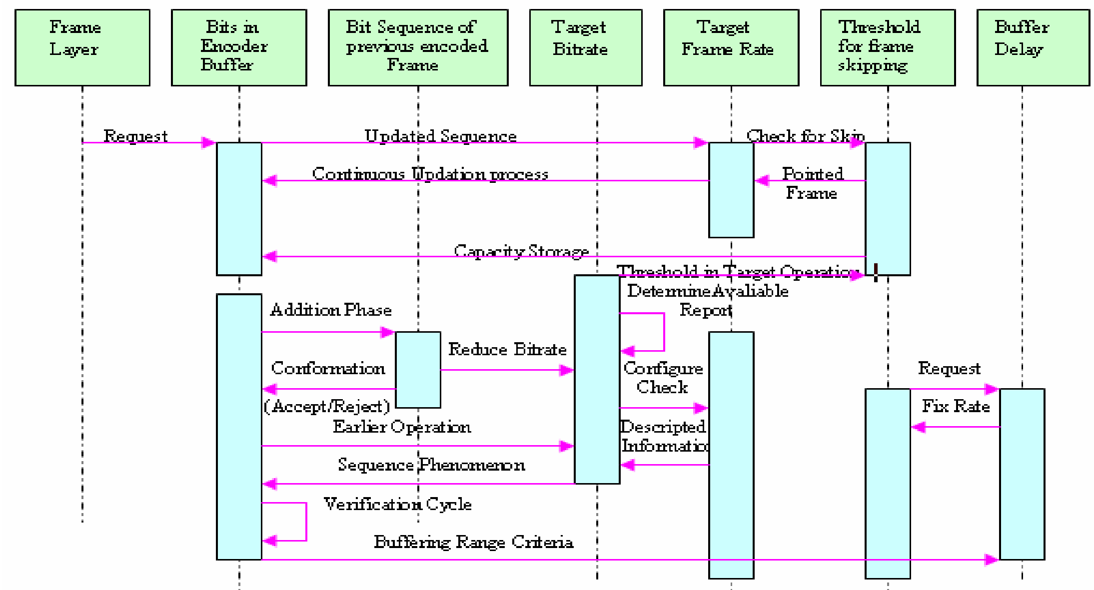

Figure 3. Frame flow control in encoded buffer

intraDC is performed by using the following formulas:

|REC| = QP * (2 * |LEVEL| + 1) QP => odd
|REC| = QP * (2 * |LEVEL| + 1) – 1 QP => even

To control the sign property of the REC after calculation, the following equation is used.

REC = sign (LEVEL) * | REC |

E. Entropy Coding

Entropy coding operates using the variable length code (VLC) having the Huffman coding algorithm to give the optimal bitstream code with minimum redundancy. Ignoring the binary fraction and concentrate on the discrete number of bits are severely effected the overall accuracy and performance of the compression ratio. To achieve better compression, syntax-based arithmetic coding executes their operation by taking the extreme (H, L) ranges of data and find the subranges for all data symbols fall in the extreme range. Larger amount of interval contain less amount of fractional bitstream which produces shorter codebooks. Finally, fractional bitstream output is achieved and transferred in a sequence (FIFO) manner for buffering to improve data flow efficiency.

F. Buffering activity with Rate Control

Observing the behavior of the encoded buffer when the real-time video is fully stored the buffer, so rate control may either be skip the frames or reduce bit rate. Rate Control maintains the buffer delay by considering the frames to set the number of bits for every operated frame.

Figure 3 shows the maintenance of buffering delay by considering two cases in which one show the skipping process having the bitrate in encoder buffer configure with the threshold of skipped frame. Otherwise, target frame moves towards the next frame in the original video sequence indicating the continuous flow.

3. Real-Time Video Decoding of H.263

Receiving the compressed bitstream in a sequence order, the entropy decoder performs function in two different ways. First is the macroblocks (INTRA/INTER) which take the coefficient values and the other is motion vector which moves towards the motion compensation by using the advanced prediction strategy.

Reconstruction of the original macroblocks acquired from the INTRA/INTER macroblocks by decoding the bitstream to get the quantized transform coefficients which passes through the inverse quantization. Macroblocks of the reconstructed frames behave as the decoded image and also stored in the frame storage buffer which is used as the output reconstructed frames(video sequences). Process diagram of video decoder is shown in Figure 4.

Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06)
0-7695-2560-1/06 $20.00 © 2006 IEEE

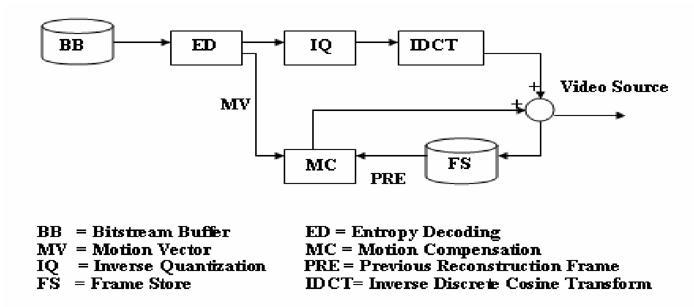

Figure 4. Data Flow Structure of the H.263 Video decoder

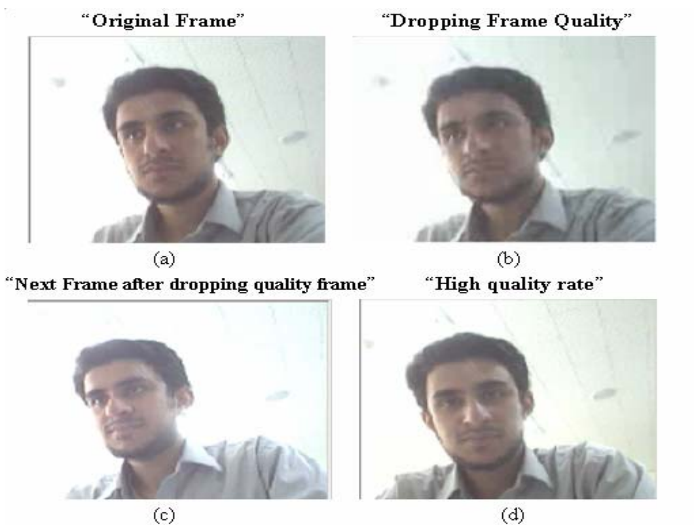

Figure 5. (a) shows the original frame of "16" frame number, while (b) shows the downfall regarded with quality rate. (c) indicate the next frame after dropping quality frame. Finally, (d) show the normal PSNR at decoding rate

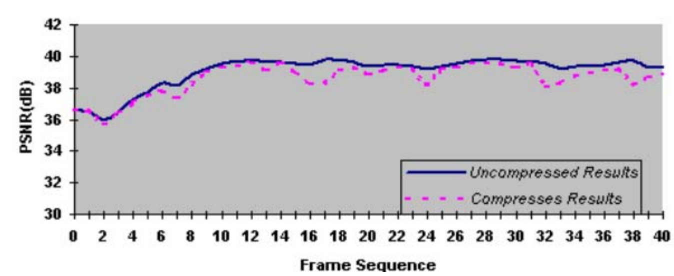

Figure 6. Comparison of frame quality (PSNR) between the original and compressed frame

4. Overall Performance

To analyze the performance of the overall operations which cause to reduce the complexity period without affecting the video quality. Our system has frame sequence containing a frame-rate of 30fps at 25kbps, but the initial results are encoded with a frame rate of 10fps. Video sequences contain the body parts movement with a still background. Due to flow control of bitstream at the stage of buffer filling process, frame skipping rate is 4 out of first 40 frames in the retransmission process.

Figure 5 represents the brief quality phenomenon in real-time video codec. Figure 5(a) show the originality of the frame while on the other hand (b) configure a dropping quality result of about 37.82dB. Showing the behavior of the next frame after dropping frame containing frame number of "18", maintain their quality without representing distortion movement. The performance of video quality is shown in Figure 6.

The experimental work shows that dropping of PSNR grading at some occasions has not prejudicially affected the video quality due to a uniform approach while their average difference is 0.31dB which indicates some significant achievement.

Figure 7 configure the bit/frame by considering the both proposed features comparison.

Experimental results show some up/down flow movement in bitrate throughout of the video sequence, while our results consist of first 80 frames with an average bitrate of 24.92kb/s having 10f/s. The H.263 codec (tmn8) containing some modified operations are compared with Telenor Research software codec and CHC encoder.

Table 1 performed their operation in QCIF environment which shows some closer results in compression mechanism while better achievement in case of video quality.

Figure 8 shows the practical implemented results containing original video which is encoded and stored in buffer containing bitstream form then decoded video as output in a real-time environment.

5. Conclusion

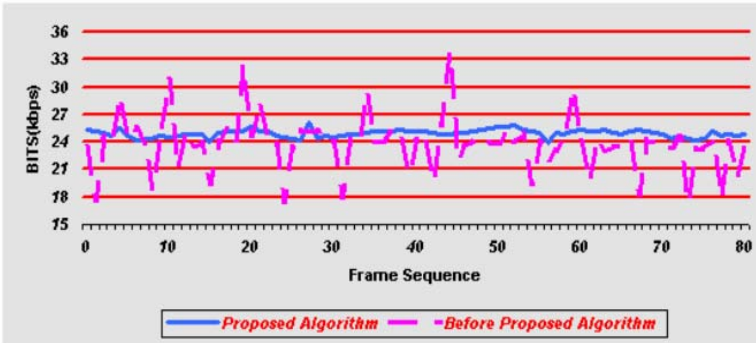

Figure 7. Comparison of the number of bit/frame used by TMN8 software and after applying proposed algorithm in the H.263 codec

Table 1. Compression ratio and quality PSNR (dB) results of H.263 Codec, Telenor developed codec and CHC encoder [8].

Description	Uncompressed Bitrate(Mbit/s)	Frame/Sec	Average PSNR(dB)	Bitrate (Kbit/s)	Compression Ratio
H.263Codec	**3.0**	**10**	**39.27**	**24.92**	**120:1**
Telenor Research S/f (Case 1)	3.0	10	38.51	22.81	133:1
Telenor Research S/f (Case 2)	3.0	10	41.75	56.70	54:1
CHC Encoder	4.56	15	31.45	46	99.13:1

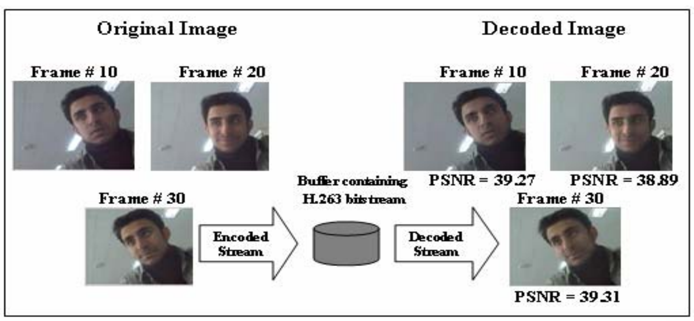

Figure 8. Example of Real-time H.263 video codec

In this paper, we justify to enhance the performance quality with low-bitrate video communication in the real-time video environment. Our intension is to speed up and concentrate the blocking activities by considering the formulated criteria to control the distortion measure with searching method. The proposed phenomenon configures a uniform reduction of computational complexity of DCT and entropy coding with a limited influence of compression efficiency. While performance results show the PSNR scaling in a safe mode which provides positive achievement of video quality and precisely describes the narrow activities which plays role in performance efficiency. Future research may reduce the PSNR difference of video quality and improvement in compression efficiency. The proposed system can be used for unidirectional applications such as a remote surveillance system and may be also used for interactive applications such as video conferencing remote lecture.

6. Acknowledgement

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

7. Reference

[1] I. E. G. Richardson, and Y. Zhan, "Video Codec Complexity Management," Proc. Picture Coding Symposium, Seoul, Korea, Apr. 2001.
[2] A. Jalal, and S. Kim, "The Mechanism of Edge Detection using the Block Matching Criteria for the Motion Estimation," Proc. Human Computer Interaction, pp.484-489, Jan. 2005.
[3] http://www-mobile.ecs.soton.ac.uk/peter/h263/h263.html
[4] M. Alkanhal, D. Turage, and T. Chen, "Correlation based search Algorithms for motion estimation," Picture Coding Symposium, Portland, Oregon, Apr. 1999.
[5] J. Liang, and T. D. Tran, "Fast multiplierless approximation of the DCT with the lifting scheme," IEEE Trans. on Signal Processing, vol. 49, no. 12, pp. 3032-3044, Dec. 2001.
[6] Video Coding Experts Group, ITU, Video codec test model, TMN8, 1997.
[7] N. Kamaci, and Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards," IEEE Int. Conf. on Multimedia and Expo, vol.1, pp.345-348, July 2003.
[8] S. Aramvith, I. Pao, and M. Sun, "A Rate-Control Scheme for Video Transport over Wireless Channels," IEEE Trans. Circuits Sys. Video Tech., pp.569-280, May 2001.
[9] C. Derviaux, F. X. Coudoux, M.G. Gazalet, and P. Corlay, "Blocking artifact reduction of DCT coded image sequences using a visually adaptive postprocessing," Proc. IEEE Int. Conf. on Image Proc., Lausanne, Sept. 1996.
[10] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of Blocking Artifacts in Image and Video Coding," IEEE Trans. Circuits Sys. Video Tech., vol.9, no.3, pp.490-500, Apr. 1999.

0-7695-2560-1/06 $20.00 © 2006 IEEE