MOTION WAVELET DIFFERENCE REDUCTION (MWDR) VIDEO CODEC

Yee L. Law and Truong Q. Nguyen

Department of Electrical & Computer Engineering
University of California, San Diego
Email: ylaw@ucsd.edu, nguyent@ece.ucsd.edu

ABSTRACT

In this paper, a new fast video codec using Wavelet Difference Reduction (WDR) algorithm is presented. This proposed video codec is inspired by the concept of motion JPEG; namely, we adapted the efficient WDR still image compression algorithm into a video compression algorithm without the motion estimation step. This approach significantly reduces the processing time for motion vector search and motion compensation procedures. In addition, we employ block-based WDR algorithm which significantly reduces the memory requirement while comparing to other wavelet based compression algorithm.

1. INTRODUCTION

With the explosive growth of the wireless communication technology, video conferencing over the mobile phone or PDA becomes increasingly demanded. However, bandwidth limitation is still a major constraint on video mobile phone delivery and service.

There are several existing well-known standard video codecs (Motion JPEG [1, 2], MPEG [3, 4], and H.26x [5, 6]). At the high end, ITU-T video coding standard, H.26x, has been shown to provide superior coding efficiency for video conferencing applications. However, the trade-off for this high video quality is high complexity which can not be implemented in small devices, e.g. mobile phone or PDA. On the other hand, motion JPEG [7] is a very simple motion pictures codec with good video quality at high bit rate.

In this paper, we propose a mid-point solution – simple codec with good video quality at low bit rate. This work is motivated by the idea of extending an efficient still image codec into a moving picture codec. The Wavelet Difference Reduction algorithm [8] is a simple and efficient coding method; therefore, this coding technique is used in our video codec (we refer as motion WDR or MWDR).

The most complicated part in video coding design is motion estimation which exhausts most of the computing power in searching for the correct motion vectors. Hence, motion estimation is avoided in this proposed algorithm. The details are explained in the following sections. The experimental results show that even without those precise motion vectors, MWDR still can provide good visual quality for video conferencing in low bit-rate situation.

This work is supported by office of Naval Research.

2. OVERVIEW OF WDR

In WDR compression algorithm, discrete wavelet transform (DWT) is applied on the input image. Then bit stream is coded based on the significant coefficients of the bit-plane (from Most-significant-bit to Least-significant-bit) [9]. The WDR algorithm consists of four parts, as shown in Fig. 1. In the Initialize part, an initial threshold value T is chosen so that all transform values have magnitudes less than T, and at least one has magnitude greater than or equal to T/2. The purpose of the loop indicated in Figure 1 is to encode significant transform values by the method of *bit-plane encoding*. A binary expansion, relative to the quantity T, is computed for each transform value. The loop constitutes the procedure by which these binary expansions are calculated. As the threshold is successively reduced by half, the Significance Pass and Refinement Pass compute the next bit in the binary expansions of the transform values. See [3, 9] for a more complete description of *bit-plane encoding*.

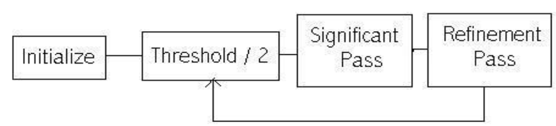

Figure 1: Wavelet Difference Reduction Algorithm

Bear in mind that *bit-plan encoding* encodes the distances between significant coefficients. For highly correlated neighboring pixels, the distances between them are very close and result in efficient encoding. In contrast, scatter coefficients require more bits to code due to large distances between significant coefficients.

To realize the maximum benefits of WDR (fast implementation and low power requirement), Daub 5/3 Int-to-Int transform [10] is used. For most wavelet-based codec, significantly large amount of memory is required in order to store the wavelet transform of the image. Thus a block-based

WDR is implemented which divides the transform into small blocks and each block is loaded into the memory individually for coding processes. This extensively reduces the internal memory and power requirements.

3. DESCRIPTION OF MWDR

The main idea of MWDR is to convert a still image codec into a moving picture algorithm. Let us first discuss the MWDR syntax hierarchy. We use a hierarchy that is very simpler as H.263. Sequences are group into group of pictures (GOP) and each frame consists of luminance (Y) and chrominance (UV) information. Since block-based WDR is used, the smallest process unit is block. For QCIF resolution, the Y frame is partitioned into 256 blocks and UV frame, which is a quarter size of Y, is partitioned into 64 blocks for block-based processing.

WDR coding method is chosen because of its simplicity and efficiency. In order to retain the simplicity of WDR, only I and forward predicted (inter/P) picture is used and the I/P pictures are organized into GOP. As mentioned before, no motion estimation is utilized in MWDR. One may wonder how the P picture is constructed. The P picture is calculated as the difference between the previous and current frames in block-based manner. Therefore the P picture can be viewed as error frames. In video conferencing application, sequences mainly contain talking head with constant background, these error frames consist large amount of insignificant values (constant background) and partial large values (head, eyes or mouth movement). Therefore, the error frames can be efficiently coded.

3.1. Encoder

Figure 2 depicts the system diagram of the encoder. Due to varying contents of different type of pictures (I/P), different levels of decomposition are applied to different pictures type. Consequently, the encoder first needs to decide the type of the current frame according to the pre-defined GOP order. For I picture, a 4-level wavelet decomposition is used on Y frame and 3-level decomposition is applied on UV frame. For the reason that I frame contains high energy information and due to the wavelet transform nature, these high energy coefficients will be closely packed in high level decomposition, namely the LL band. As mentioned before, WDR performs more efficient in a compact environment. Therefore, high level decomposition is more desirable for I picture encoding. On the other hand, P picture (for all YUV frame) is decomposed in 1-level due to its low information structure. Small magnitude will be spread all over the four levels and this greatly reduces the WDR coding efficiency.

After the transformation, I picture is sent to the WDR encoding processing immediately. For P picture, motion is allocated by comparing the sum absolute difference (SAD) of each block to a pre-defined threshold. This thresholding procedure only operates on Y frame since the changes should be appear at corresponding locations of UV. Therefore, once the major movement locations are found on Y frame, the corresponding remapped motion locations can be tracked on UV.

$$SAD = \sum_{i=1}^{N} \sum_{j=1}^{M} \left| p_{ij} - \hat{p}_{ij} \right|$$

where p_{ij} and $\hat{p}_{ij}$ are the current and previous ij^{th} pixel values respectively. The SAD and thresholding are done in block-based matter. Therefore, N and M are the dimension of the block. The purpose of SAD computation is to locate the changes between frames and since not all motions are noticeable by human eyes, threshold comparison is employed to indicate the significant motion blocks which will be sent for encoding.

While all the blocks are compared and a difference frame is formed, the rate control module will estimate the right amount of bits for the difference frame to be efficiently coded. The rate control method is described in the later sub-section.

Similar to many video codecs, a decoder is implemented at the encoder side for the purpose of decoding the current bitstream and storing the reconstructed frame in buffer for motion estimation on the next frame.

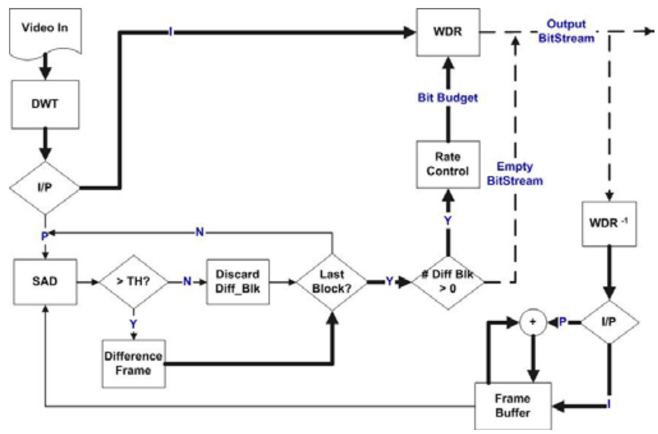

Figure 2: Encoder Diagram
(Thick line = frame process; thin line = block process; dotted line = bitstream)

3.2. Decoder

The decoder which basically carries out reverse steps as the encoder. First, compressed bit-stream is decoded by inverse WDR. For P picture, a previously reconstructed frame is added. Then the decoded frame is reconstructed using inverse discrete wavelet transform.

Since the inter frames are coded with low bit-rate, the information of the background is mostly lost. The background intensity cannot be maintained throughout the sequence and this causes blanking effect in the compressed sequences. In order to preserve the background information without using higher bit-rate, the prior background information is employed to adjust the currently background. This is done in both encoder and decoder where the addition of the reconstructed P frame and previous frame takes place. Due to the constant background assumption, most of the background information appears in the LL subband of the wavelet transform. Consequently, the difference between the average values of the LL subband of the two reconstructed frames are computed. This difference average value is the missing information which is compensated to the currently encoded or decoded frame.

3.3 Rate Control

The rate control process in the recent standard video compression algorithm, H.26x or MPEG, is highly dependent on the quantization steps. However, we have no control over the quantization steps in the proposed codec. Hence, those commonly used rate control methods in the standard are not applicable to MWDR.

We performed extensive experiments and noticed a linear relation between bit per pixel (bpp) and number of significant coefficients. Figure 5 shows an example of bpp vs. number of significant coefficient that need to be coded in a frame. This linear relation is employed and the simulation results show that this rate control method is very efficient.

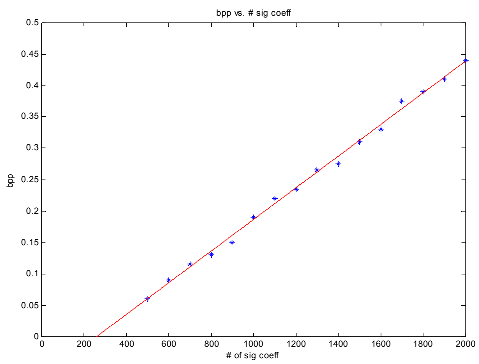

Figure 5: number of significant coefficients and bit rate relationship

4. EXPERIMENTAL RESULTS

In the simulation, three test image sequences are used: Grandma, Salesman, and Claire. Since this proposed codec targets at video conferencing application, these three test sequences contain different degrees of head movements which can fully evaluate its performance. The sequences are in QCIF format with frame size 176 × 144 and in YUV raw data format. The structure of each GOP is chosen to be one I frame followed by eleven P frames and frame rate is set to 25 fps.

In order to demonstrate the performance of the proposed algorithm, simulation includes two sets of comparisons and PSNR is used as quality measurement. First set is the comparison with low complexity video codecs such as MJPEG and WDR and second set is comparison with high complexity video codecs, H.263 [11] and H.264 [12].

For the first set of comparison, the test sequences are compressed at 250 kbps. Table 1 summarizes the average PSNR. Since MJPEG is a very simple codec which does not utilize any motion estimation and compensation method, it is expected that MWDR will outperform it significantly in all cases. Indeed the reconstructed sequences are suffering severe block artifacts and quantization error. For WDR, bits are equally distributed among all frames and relatively each frame obtains small among of bits, which are not sufficient to code the detail and this causes blurring artifact (see simulation results at http://videoprocessing.ucsd.edu/people/louise).

The second set of comparison includes MWDR, H.263, and H.264. In this set, two different bit rates (70kbps and 150 kbps) are evaluated. Tables 2-3 show the average PSNR and due to the limited of space, only visual quality results of 70 kbps is shown (Figure 6). The simulation results for 150 kbps are available at the URL mentioned previously. In the comparison between MWDR and H.263, MWDR outperforms H.263 by 0.1-1dB at low bit rate situation (70 kbps). At high bit rate, H.263 outperforms MWDR by 2 dB for Grandma and Claire sequence; however, they have comparable visual quality. In both PSNR and visual quality comparison, H.264 achieves the best result among the three codecs. This result is well expected since H.264 is the recent leading video codec. However, it is a very complex codec which requires long computation time.

	MWDR	WDR	MJPEG
Grandma	37.6	32.29	29.96
Salesman	36.7	30.95	28.98
Claire	37.71	34.64	30.04

Table 1: Average PSNR (dB) at 250 kbps

	MWDR	H.263	H.264
Grandma	33.66	33.76	36.98
Salesman	33.62	32.56	35.95
Claire	36.18	35.51	42.91

Table 2: Average PSNR (dB) at 70kbps

	MWDR	H.263	H.264
Grandma	36.54	38.65	42.01
Salesman	35.89	35.13	39.82
Claire	37.36	40.01	46.96

Table 3: Average PSNR (dB) at 150kbps

Besides PSNR and visual quality comparison, a brief algorithm complexity comparison is summarized in this section. In MJPEG, all frames are coded as I picture type which makes the codec extremely simple. Therefore, we only compare the complexity between MWDR and H.263 codecs. It is fair to assume both codecs offer similar I picture compression complexity because discrete wavelet transform and discrete cosine transform require similar operation processes. The most complicate part in coding P picture is motion estimation procedures; thus, complexity is measured by the number of operation needed in motion vector search. In the following, the dimension of macro block is assumed to be $M \times M$ and MSE comparison is used.

For H.263, there is a variety of motion vector search algorithms. The complexity comparison on full search (the most common one) and diamond search is presented. The average number of search points per motion vector generation is provided in [13]. MSE calculation requires M^2 times of subtraction and M^2 times of multiplication. For full search, 1024 search points (typically 32×32 search window) are required, that brings the total number of operations equals to 2048 M^2. Likewise, diamond search needs average of 15 search points which takes 30 M^2 times of operations to compute one motion vector. In MWDR, motion is predicted by the error frame which is computed in one simple subtraction step. For a $M \times M$ block, the total number of operations equal to M^2 .

5. CONCLUSION

In this paper, a new efficient motion picture codec is proposed. The motivation of this codec is to develop a simple and fast video codec for low bit rate communication application. MWDR is built upon the still image codec WDR algorithm with the proposed rate control method. Performance comparison shows MWDR outperforms MJPEG significantly. Although H.263 provides better PSNR results than MWDR at high bit rate; their coding efficiencies are comparable at higher frame rates. The visual qualities of both codecs are comparable. As expected, H.264 provides the best quality among all codecs. However, MWDR is a much simpler codec than H.26x with significant lower implementation complexity and cost.

6. REFERENCES

[1] G.K. Wallace, "Overview of the JPEG (ISO/CCITT) still image compression standard," *Image Processing Algorithm and Techniques, Proc. of the SPIE,*, pp 220-233, Feb. 1990.

[2] ISO/IEC 10918-1 /ITU-T Recommendation T.81, "Digital compression and coding of continuous-tone still image," 1992 [JPEG].

[3] J. Watkinson, *The MPEG handbook: MPEG-1, MPEG-2, MPEG-4*, Focal Press, Boston, 2001.

[4] http://www.cselt.it/mpeg/ [MPEG committee official site].

[5] ITU-T Recommendation H.263 Version II, "Video coding for low bitrate communication," Nov. 1998.

[6] Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*, John Wiley, England , 2003.

[7] W.B. Pennebaker, and J.L. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[8] T.Q. Nguyen, and J.S. Walker, Wavelet-based image compression, *In "Handbook of Transforms and Data Compression,"* K.R. Rao, and P.C. Yip (Eds.), CRC Press, Boca Raton, 2001.

[9] J.Tian and R. O. Wells, Jr. "A Lossy Image Codec Based On Index Coding," *IEEE Data Compression Conference,* DCC'96, page 456, 1996.

[10] A.R. Calderbank, I. Daubechies, W. Sweldens, and B.L. Yeo, Wavelet Transforms That Map Integers To Integers. *Applied and Comp. Harmonic Analysis*, 5, 332-369, 1998.

[11] K.Jack, *A Handbook for the Digital Engineer: Video Demystified*, LLH Technology Publishing, Eagle Rock, VA, 2001.

[12] J.Ostemrmann, J.Bormans, P.List, D.Marpe, M.Narroschke, F.Pereira, T.Stockhammer, and T.Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, Vol. 4, Issue 1, page 7-28, 2004.

[13] Y. Nie, and K.K Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEE Transactions On Image Processing*, Vol. 11, No. 12, December, 2002.

(a) Original (b) MWDR
(c) H.263 (d) H.264
(a) Original (b) MWDR
(c) H.263 (d) H.264
(a) Original (d) MWDR
(c) H.263 (d) H.264

Figure 6: MWDR, H.263, H.264 comparison at 70kbps