On Scalable Lossless Video Coding Based on Sub-Pixel Accurate MCTF

CIPR Technical Report TR-2006-8

Sehoon Yea and William A. Pearlman

January 2006

Center for Image Processing Research

Rensselaer Polytechnic Institute
Troy, New York 12180-3590
http://www.cipr.rpi.edu

On scalable lossless video coding based on sub-pixel accurate MCTF

Sehoon Yea[a] and William A.Pearlman[a]

[a]Center for Image Processing Research
Electrical, Computer & Systems Engineering Dept.
Rensselaer Polytechnic Institute
110 8th St. Troy, NY 12180-3590

ABSTRACT

We propose two approaches to scalable lossless coding of motion video. They achieve SNR-scalable bitstream up to lossless reconstruction based upon the *subpixel-accurate MCTF-based* wavelet video coding. The first approach is based upon a two-stage encoding strategy where a lossy reconstruction layer is augmented by a following residual layer in order to obtain (nearly) lossless reconstruction. The key advantages of our approach include an 'on-the-fly' determination of bit budget distribution between the lossy and the residual layers, freedom to use almost any progressive lossy video coding scheme as the first layer and an added feature of near-lossless compression. The second approach capitalizes on the fact that we can maintain the invertibility of MCTF with an arbitrary sub-pixel accuracy *even in the presence of an extra truncation step for lossless reconstruction* thanks to the lifting implementation. Experimental results show that the proposed schemes achieve compression ratios not obtainable by intra-frame coders such as Motion JPEG-2000 thanks to their inter-frame coding nature. Also they are shown to outperform the state-of-the-art non-scalable inter-frame coder H.264 (JM) lossless mode, with the added benefit of bitstream embeddedness.

Keywords: lossless video compression, MCTF, lifting scheme, scalable video coding, sub-pixel accuracy

1. INTRODUCTION

The ongoing efforts in the wavelet-based Scalable Video Coding (SVC) technology and its recent merge into the H.264 JVT/AVC standard reflect the current real-world needs for scalable dissemination of video data.[14, 16] On one extreme side of such scalability needs is lossless reconstruction of video. Although lossless compression of data with more of a static nature — such as digital image and text — achieved a certain level of success, its motion-video counterpart has been among the far less explored territories in the coding area possibly because of the lack of needs and technologies that would make it practically feasible. However, with the advent of newly emerging archival applications such as digital cinema and surveillance, its potential utility/necessity is already a reality, as evidenced by recent adoption of Motion JPEG-2000 by DCI (Digital Cinema Initiative) and some of the MPEG activities.[9, 14–16] A Majority of the lossless video compression schemes in the literature are predictive-coding based, hence either non-scalable or scalable in a limited sense(e.g. FGS).[8, 10–13] On the other hand, intra-frame coding methods based on integer-wavelet transforms such as Motion JPEG-2000 obviously generate SNR-scalable bitstreams and they allow a faster data retrieval which could be a big advantage for applications like video editing. However, there could be many scenarios where higher compression efficiency achieved through inter-frame encoding is preferred over the random accessibility of video content.

When it is necessary to provide SNR-scalability as well as temporal and resolution scalabilities, MCTF (Motion-Compensated Temporal Filtering)-based SVC is an attractive option.[18, 19] However, how to design such a video coder especially with *SNR-scalability up to lossless reconstruction* has not been often discussed in the literature. In this paper, we present some of the results of two possible approaches to fulfill such a design goal. In particular, we want to focus on incorporating sub-pixel accurate motion compensation for effective lossless inter-frame video coding. Unlike prediction-based coders such as MPEG-2, where interpolation of pixel values for sub-pixel accurate motion-compensation can be performed both at the encoder and the decoder based

Send correspondence to William A. Pearlman : E-mail: pearlw@ecse.rpi.edu, Telephone: 1 518 276 6082

Visual Communications and Image Processing 2006, edited by John G. Apostolopoulos, Amir Said,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6077, 60771D, © 2005 SPIE-IS&T · 0277-786X/05/$15

on the identically reconstructed frame data, MCTF-based coders have had a motion invertibility problem with sub-pixel accuracy which has only recently been resolved via lifting-based implementations.[18,19] When it comes to lossless compression, in the predictive-coding approach, simply truncating interpolated pixel-values at sub-pixel points (real numbers, in general) to nearby integers and losslessly compressing the resulting integer-valued residual will guarantee lossless —although not scalable— reconstruction of the video.[12] However, to the best of authors' knowledge, few attempts have been made in the literature to achieve scalable lossless reconstruction using MCTF-based wavelet video coders*. In this paper, we demonstrate two approaches to achieve the aforementioned design goal. One is an extension of the two-stage coding method we proposed previously. The other is called *LS(Lossless)-MCTF* and it is a reversible version of the MCTF for lossless reconstruction. Both can enjoy the benefits of sub-pixel accurate motion compensation for high compression ratios and each of them has its own distinctive features as we will discuss in later sections.

The organization of the paper is as follows. Section 2 describes the two-stage approach. Then the LS-MCTF approach is introduced and examples of the motion compensated S and 5/3 transforms are presented in Section 3. We then present some of the experimental results in Section 4 and conclude the paper in Section 5.

2. APPROACH I : MOTION-COMPENSATED TWO-STAGE NEAR-LOSSLESS CODER

The first method is based on the motion-compensated version of the two-stage near-lossless coder previously proposed by the authors, where a wavelet-based coder such as SPIHT in the first encoding stage is followed by a simple adaptive arithmetic coder for residual coding.[1,2]

2.1. A Wavelet-Based Two-Stage Near-Lossless Coder

Figure 1 shows a schematic block diagram of the two-stage near-lossless coder we used in this work. In the figure, we assume any (bi)orthogonal wavelet encoder based on successive bit-plane encoding. At the encoder, a lossy reconstruction is subtracted from the original input image to form a residual. Noting that $\|e - e_{dec}\| \leq \delta$ is equivalent to $\|I - I_{near-lossless}\| \leq \delta$, the residual e is uniformly quantized as follows to guarantee the maximum error no larger than δ:

$$\begin{aligned} e_{q-idx} &= \lfloor \frac{e+\delta}{2\delta+1} \rfloor, \quad e > 0 \\ e_{q-idx} &= \lfloor \frac{e-\delta}{2\delta+1} \rfloor, \quad e < 0 \end{aligned} \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes the integer part of the argument. The generated quantization index e_{q-idx} is then losslessly encoded by a simple arithmetic coder without incorporating any context model.

At the decoder, we decode e_{enc} by arithmetic decoding to yield e_{q-idx} followed by a dequantizer defined as follows in order to guarantee $\|e - e_{dec}\| \leq \delta$.

$$e_{dec} = (2\delta + 1) \cdot e_{q-idx} \tag{2}$$

By adding the lossy reconstruction I_{lossy} and the dequantized residual e_{dec}, we obtain the final near-lossless reconstruction $I_{near-lossles}$ which guarantees $\|I - I_{near-lossless}\| \leq \delta$.

Now, in a typical two-stage approach like this, efficient encoding of the given residual is the key issue, where, for example, we could build an efficient context model exploiting the correlation between the lossy reconstruction and the residual[3] or perform a search for a wavelet packet basis for the residual.[4] However, if we want the total (lossy plus residual) bit-rate to be minimized for a given δ, then such approaches will not be efficient to achieve this goal because one usually has to resort to exhaustive iteration to find out the lossy layer bit rate that would minimize the total bit rate.

*At the time of writing this paper, the authors became aware of such an attempt by A. Secker and D. Taubman in the context of scalable motion coding[24](see Section V-A). However, their focus was not on lossless compression and no comparison with other methods was made.

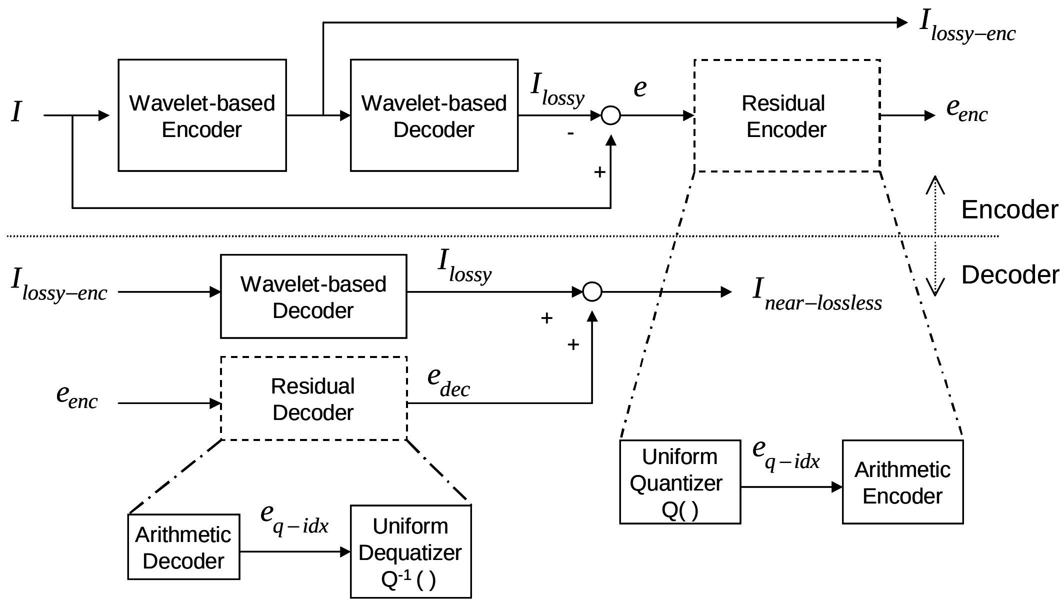

Figure 1. Two-Stage Near-Lossless Wavelet Coder

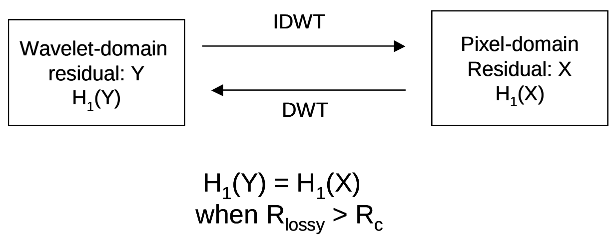

Figure 2. First-Order Entropies in Two Domains

An efficient 'on-the-fly' algorithm was proposed in our previous work[1,2] to find such an optimal lossy layer bit rate while coding the source only once without any iteration. Figure 2 shows the central idea of our method to determine the optimal first-stage rate. When the bit rate (R) for the lossy reconstruction (I_{lossy}) becomes larger than the 'critical rate' (R_c), the encoding residual of a source becomes white,[5,6] and the residual and its wavelet transform converge to each other in probability distribution.[1] Therefore, the first-order entropy $H_1(Y)$ of the wavelet transform residual and the first-order entropy of its inverse, i.e. the source residual's entropy $H_1(X)$ are equal.[7] On the other hand, beyond R_c, the white (uncorrelated) residual lacks the structure that a good lossy encoder (e.g., SPIHT) can take advantage of, so the coding efficiency of such a coder tends to become at most as good as and usually worse than that of first-order entropy coding. Therefore, assuming a first-order encoding of the source (i.e. pixel-domain) residual by the arithmetic coder, the total bit rate (lossy layer bit rate R plus the first-order entropy of the residual $H_1(\cdot)$) versus lossy layer bit rate curve has a point or a small flat region of bit rates where it attains its minimum. These observations also apply to near-lossless cases where the residual is quantized as (1)[1] and form the bases of the near-lossless coding scheme we proposed.

Another thing important to realize is that beyond such a critical rate, the aforementioned total bit rate versus lossy layer bit rate curves in both domains converge due to the fact that $H_1(Y)$ and $H_1(X)$ are equal. Hence, we only need to know the first-order entropy ($H_1(Y)$) of the wavelet-domain residual in order to estimate the actual total bit rate ($R + H_1(X)$). This means we do not need to generate lossy reconstructions at various bit rates and take the inverse wavelet transform to generate the quantized pixel-domain residuals at corresponding bit rates. Potential savings in computation obtained thereby can turn out to be significant especially when dealing with 3D sequences.

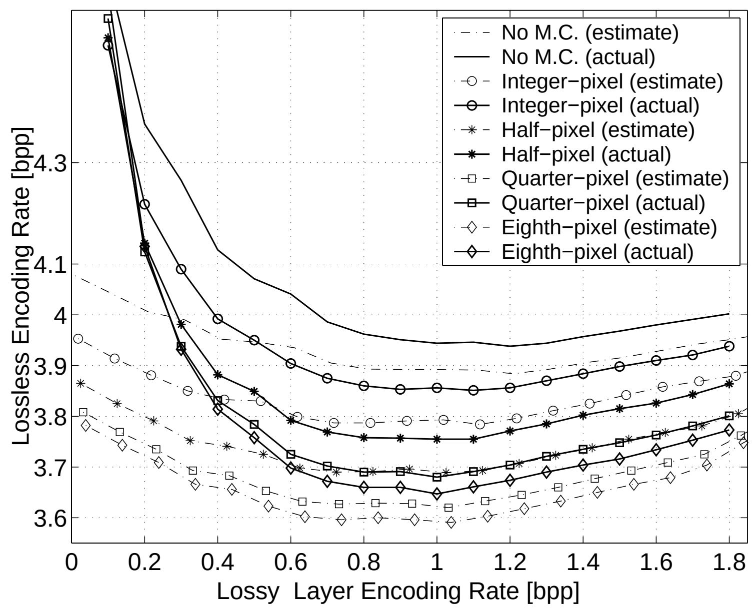

Figure 3. On-the-fly Estimation of Total Bit Rate in Approach I (*Foreman* CIF frame: 122 - 137 , Y only)

2.2. Two-Stage Motion-Compensated Scalable Lossless Coder

Notice that lossless compression is nothing but the special case with δ=0 in (1) and (2). Therefore, by equipping the 'wavelet-based encoder' with a motion compensation routine, we can construct a two-stage motion-compensated lossless coder by setting δ=0. We say the coder is 'scalable' in the sense that the first-stage coder (e.g. a motion-compensated 3D SPIHT) will naturally provide embedded bitstream in the MSE sense up to the point where we switch to the pixel-domain residual encoder for near-lossless reconstruction, as described in Section 2.1. Figure 3 shows an example of the 'on-the-fly' estimation with the two-stage method for the luminance frames of *Foreman* sequence (CIF, frames 0 - 15) with various sub-pixel accuracy motion-compensations. In the figure, the 'actual' curves plot the actual lossless bit rates which were obtained by adding to the lossy layer bit rates the second-stage encoding rates obtained by repeatedly coding the residuals generated at various lossy layer bit rates. On the other hand, the 'estimate' curves are obtained 'on-the-fly' inside 3D-SPIHT while encoding the source only once, with a minimal overhead in calculation. When the 'estimated' total rate starts to increase around 1 bpp as in the figure, 3D-SPIHT can stop encoding and switch to the pixel-domain arithmetic coder for the residual encoding, thereby achieving the minimum lossless bit rate without iteration, saving potentially significant amount of computation. With this approach, users can enjoy SNR-scalability up to the optimal lossy bit rate (usually achieving around 45 dB), with a following residual layer to make the final reconstruction lossless.[1,3] The two-stage strategy not only allows to incorporate sub-pixel accurate motion-compensation naturally, but also retains most benefits of typical MCTF-based (wavelet) video coders such as uncompromised PSNR's at low-to-medium bit rates often hard to achieve with integer-wavelet based coders.

3. APPROACH II : LOSSLESS-MCTF WITH SUB-PIXEL ACCURATE MOTION-COMPENSATION

The second method is based on the newly-proposed LS (Lossless)-MCTF which is a lifting-implementation of the motion-compensated wavelet-transform for temporal de-correlation of video sequences along the motion trajectory. To demonstrate the idea, we first show a lifting implementation of the motion-compensated S-transform to make the resulting residual frame data integer, which could be efficiently compressed using integer-to-integer spatial wavelet coders such as S+P.[21] As is well-known in the literature,[18,19] transform invertibility is maintained with an arbitrary sub-pixel accurate motion interpolation thanks to the lifting structure. However, note that what we need here is the invertibility of the lifting transform in the absence of quantization error not with

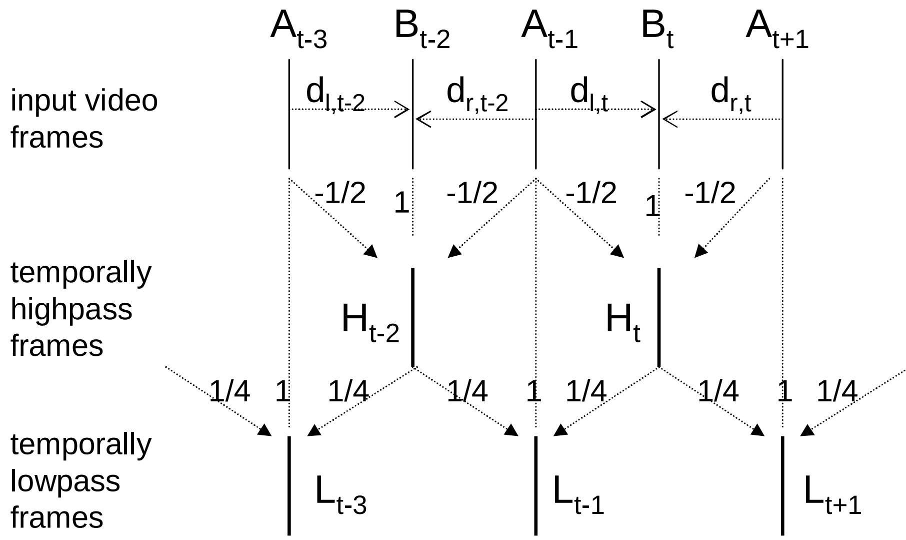

Figure 4. MCTF using biorthogonal 5/3 filters with bidirectional MC

high-precision floating-point arithmetic, but with integer one. Fortunately, incorporating any mapping including a nonlinear operation such as truncation into the prediction & update steps does not hurt the invertibility of the lifting transform.[18, 23] Therefore, we can truncate the floating-point pixel-values resulting from sub-pixel accurate spatial-interpolation in order to get the lossless reconstruction without losing the transform invertibility.

3.1. Motion-Compensated S-Transform

If we let A and B represent the reference and the predicted frame respectively and (d_m, d_n) represent a motion displacement from B to a (in general) sub-pixel position in A, then the proposed LS (Lossless)-MCTF is given as follows, where $\lfloor \cdot \rfloor$ and $\tilde{(\cdot)}$ denote downward truncation and interpolation, respectively.

At the encoder, the high (H) and low (L) pass bands are obtained by

$$H[m,n] = B[m,n] - \lfloor \tilde{A}[m-d_m, n-d_n] + 1/2 \rfloor \quad (3)$$

$$L[m,n] = A[m,n] + \lfloor \frac{\tilde{H}[m+d_m, n+d_n]}{2} \rfloor \quad (4)$$

At the decoder, the frames are reconstructed as

$$A[m,n] = L[m,n] - \lfloor \frac{\tilde{H}[m+d_m, n+d_n]}{2} \rfloor \quad (5)$$

$$B[m,n] = H[m,n] + \lfloor \tilde{A}[m-d_m, n-d_n] + 1/2 \rfloor \quad (6)$$

For the unconnected pixels in A, $L[m,n] = A[m,n]$ and $A[m,n] = L[m,n]$ are used in place of (4) and (5), respectively.[19] As mentioned above, note that truncation and interpolation in $\lfloor \frac{\tilde{H}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n]}{2} \rfloor$ in the update steps ((4) , (5)) as well as $\lfloor \tilde{A}[m-d_m, n-d_n] + 1/2 \rfloor$ in the prediction steps ((3), (6)) do not affect the invertibility of the transform in the case of sub-pixel accurate motion interpolation thanks to the lifting structure. Also note that the truncation step is necessary to make the coefficients of the resulting residual frames from the lifting filter integer, which will then be compressed by a 2D lossless scheme like S+P.[21]

3.2. Motion-Compensated 5/3 Transform

Now, it is obvious that the above idea can be easily extended to longer tap filters such as 5/3.[23] Figure 4 illustrates one stage of motion-compensated temporal analysis using the 5/3 filter. It depicts a situation where we are given an input video frames ..., $A_{t-3}, B_{t-2}, A_{t-1}, B_t, A_{t+1}$, ..., where the subscripts refer to the time and the frame type B denotes the 'predicted' frame while the type A corresponds to the 'reference' frame when the motion estimation and compensation are performed. Also the symbol $d_{x,t,m}$ with x being either l or r denotes a horizontal or vertical component of a motion vector at the horizontal or vertical position m in the predicted frame at time t pointing from the past ('l') or the future ('r') reference frame in time. The straightforward extension of LS-MCTF to the 5/3 filter case is given as follows:

For temporal analysis,

$$H_t[m,n] = B_t[m,n] - \lfloor 0.5\tilde{A}_{t-1}[m-d_{l,t,m}, n-d_{l,t,n}] + 0.5\tilde{A}_{t+1}[m-d_{r,t,m}, n-d_{r,t,n}] + 1/2 \rfloor \quad (7)$$

$$L_{t-1}[m,n] = A_{t-1}[m,n] + \lfloor 0.25\tilde{H}_{t-2}[m+d_{r,t-2,m}, n+d_{r,t-2,n}] + 0.25\tilde{H}_t[m+d_{l,t,m}, n+d_{l,t,n}] + 1/2 \rfloor \quad (8)$$

For temporal synthesis (assuming no quantization of the transform coefficients),

$$A_{t-1}[m,n] = L_{t-1}[m,n] - \lfloor 0.25\tilde{H}_{t-2}[m+d_{r,t-2,m}, n+d_{r,t-2,n}] + 0.25\tilde{H}_t[m+d_{l,t,m}, n+d_{l,t,n}] + 1/2 \rfloor \quad (9)$$

$$B_t[m,n] = H_t[m,n] + \lfloor 0.5\tilde{A}_{t-1}[m-d_{l,t,m}, n-d_{l,t,n}] + 0.5\tilde{A}_{t+1}[m-d_{r,t,m}, n-d_{r,t,n}] + 1/2 \rfloor \quad (10)$$

4. EXPERIMENTAL RESULTS

In this section, we first present the experimental results on the two proposed approaches using MC-SPIHT and compare them. Next, we implement the method proposed in Section 3.2 on MC-EZBC as an example of a wavelet-based scalable video coder with more advanced motion compensation capability to compare its lossless performance against those of the well-known state-of-the-art coders such as Motion JPEG-2000 and H.264 JM lossless.

4.1. Comparison of the Two Approaches Using MC-SPIHT

In this subsection, we compare the two proposed scalable lossless coding methods in terms of lossy and lossless performances. The first method described in Section 2 uses the MC 3D-SPIHT[20] and an adaptive arithmetic coder[26] as the first and the second stage encoders, respectively. The MC 3D-SPIHT uses the motion-compensated Haar transform along the temporal direction followed by the 9/7 floating-point filter for 2D spatial transform. The second method introduced in Section 3 uses the same MC 3D-SPIHT algorithm on the coefficients obtained by the proposed reversible motion-compensated S-transform along the temporal direction followed by the 2D S+P transforms.

4.1.1. Lossy Performance

Figure 5 shows the rate-distortion curves of the two proposed scalable lossless schemes in lossy regime on the luminance frames of the CIF sequence *Bus* with different sub-pixel motion accuracies. The curves with 'Two-Stg.' correspond to the RD curves of the base-layer (i.e. the first stage) of the aforementioned two-stage MC 3D-SPIHT. Also, 'LS-MCTF' corresponds to the second method of lossless MC 3D-SPIHT with S-transform. Use of the 9/7 floating-point coefficient filter in the former led to a better PSNR performance compared with that of the latter with integer filters, which might be further improved with a better coefficient-scaling strategy to make it closer to orthonormal.[22]

4.1.2. Lossless Performance

Table 1 compares the results of lossless compression with the two above-mentioned schemes (i.e. two-stage MC-SPIHT vs. Lossless MC 3D-SPIHT with S-Transform) on *Foreman* and *Bus* with different motion accuracies. We used only the Y components of 32 frames for each sequence. We can confirm the bit rate reduction achieved by sub-pixel accurate motion-compensation in both methods. In general, LS-MCTF performed better in lossless cases between the two proposed methods.

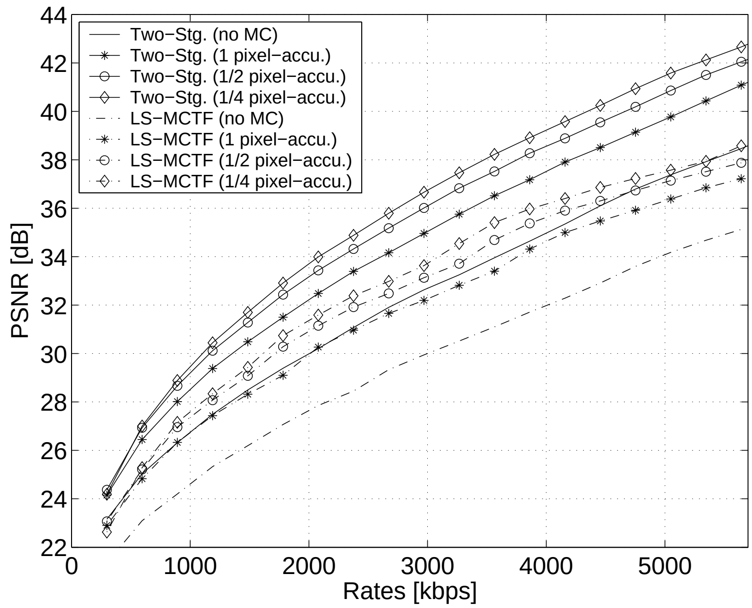

Figure 5. PSNR comparison of Approaches I & II with various sub-pixel accuracies (*Bus* CIF frame: 0-15, at 30 fps, Y only)

Sequence	foreman (frames 122-153)		bus (frames 0-31)	
Pixel accuracy	Two-Stage	LS-MCTF	Two-Stage	LS-MCTF
no MC	4.03	3.87	5.54	5.13
1	3.86 (4.2%)	3.73 (3.6%)	5.14 (7.2%)	4.76 (7.2%)
1/2	3.78 (6.2%)	3.68 (4.9%)	4.98 (10.1%)	4.63 (9.8%)
1/4	3.70(8.2%)	3.64 (5.9%)	4.87 (12.1%)	4.55 (11.3%)

Table 1. Lossless performance comparison of Approaches I & II using CIF sequences (luminance (Y) only, 32 frames only.). Numbers are in [bpp]. Numbers inside $(\cdot)$ are percentage reductions of bit rates wrt the 'no MC' case.

4.2. Comparison with Other Intra- & Inter-Frame Coders Using MC-EZBC

Since the motion estimation routine we used for MC-SPIHT was too rudimentary, it was difficult to make a meaningful assessment of the effectiveness of our schemes against other lossless video coders. This led us to implement the proposed method on MC-EZBC, which is supposed to provide a much better motion estimation and compensation capability. In order to validate the utility of the proposed scheme with a longer tap filter, we chose a version of MC-EZBC improved by RWTH, which incorporates the 5/3 motion compensated temporal filtering.[17] The proposed LS-MCTF with the 5/3 filter was implemented on it to generate the results to be presented in this section. In fact, the RWTH version of MC-EZBC incorporates several sophisticated strategies such as adaptive switching between the 5/3 and Haar filters, filter coefficient as well as the resultant transform coefficient scaling for better PSNRs. The former strategy, i.e. the adaptive switching, can be absorbed in the current framework of LS-MCTF. However, the latter, namely scaling, introduces non-integer weights and coefficients on the forward/inverse lifting steps so that the reversibility of LS-MCTF becomes difficult. Thus we had to get rid of some of such scaling stages. We leave an efficient scaling strategy for better lossy regime performance in the current LS-MCTF framework as a future work.

To assess the lossless compression performance of the proposed LS-MCTF, we chose two representative state-of-the-art coders with lossless compression functionality. As an intra-frame coder, we chose motion JPEG-2000 (kdu_v_compress.exe in Kakadu V.4.2[25]), which is nothing but a successive application of JPEG-2000 with the reversible 5/3 filter to each frame. The only coding option we needed was 'Creversible=yes' for lossless compression. Also as an inter-frame coder, H.264 lossless mode (JM ver.10.1) was used. The coding options

Encoder	MC-EZBC (5/3 LS-MCTF)			JM 10.1 Lossless	JPEG-2000
Pixel accuracy	1	1/2	1/4	RD-based(down to 1/4)	no MC
akiyo (150 frames)	1.21 (57.1%)	1.20 (57.4%)	1.20 (57.4%)	1.21 (57.1%)	2.82
bus (150 frames)	3.36 (17.4%)	3.26 (19.9%)	3.21 (21.1%)	3.73 (8.4%)	4.07
foreman (300 frames)	3.07 (16.3%)	2.98 (18.8%)	2.94 (19.9%)	3.00 (18.3%)	3.67
mobile (300 frames)	3.66 (30.2%)	3.45 (34.2%)	3.36 (35.9%)	4.09 (21.9%)	5.24

Table 2. Comparison of MC-EZBC, H.264 and JPEG-2000 for lossless compression of CIF (4:2:0 YUV) sequences. 8 bits/pel for each component. Numbers are in [bpp]. Numbers inside (·) are percentage reductions of bit rates with respect to the JPEG-2000 case

were as follows:

- Entropy coding method : CABAC
- GOP(Group Of Picture) structure : IBPBPBPBPBI
- RD optimization : enabled
- Number of reference frames : 5
- Max motion search range : 16
- Subpixel motion estimation : enabled

Notice that in the case of motion JPEG-2000, each frame can be encoded/decoded in a SNR-scalable fashion, while the JM coder does not provide such a scalability feature.

Table 2 shows the comparison results for lossless compression of four CIF (352x288) sequences (4:2:0 YUV). The lossless bit rates in the table were obtained by averaging by the total number of pixels the sum of all the bits used by the three (Y,U,V) components of all the frames and motion vectors. Also the numbers in the parentheses are the percentage reduction of bit rates with respect to the intra-frame coder— JPEG-2000.

Depending on the type of motion dominant in a given sequence, the effect of motion compensation on lossless bit rates was different. For the 'akiyo' sequence where there is almost no motion except for some of the facial movements of the news caster, bit rate reduction thanks to inter-frame encoding is huge. However, this gain seems to have been possible even without any motion compensation. This can be confirmed by looking at the almost negligible differences of bit rate reduction among different sub-pixel accuracies in the case of MC-EZBC(columns 2,3 and 4). As we can see, bit rate reduction ranged from 15% to 35% for the rest three sequences with decent amount of motion.

When compared with JM 10.1, our scheme always outperformed it sometimes significantly as shown in Table 2. This is encouraging as ours offers an added benefit of SNR-scalability. Of course there is room for improvement with the JM coder as it offers a variety of advanced coding options. However, the same can be said of the MC-EZBC for improved performance and it seems that the very fact it bypasses the transform stage for lossless reconstruction is posing a fundamental disadvantage on the JM side.[12, 13]

5. CONCLUSION

We demonstrated two ways to achieve scalable lossless coding of motion video based upon sub-pixel accurate MCTF. One was based upon a two-stage encoding strategy, and the other was a simple modification of MCTF relying upon the invertibility property of the lifting implementation. We compared the proposed method against intra/inter frame state-of-the-art coders such as H.264 lossless and JPEG 2000. The lossless compression performance was encouraging, not to mention the obvious advantage of SNR scalability of the proposed approaches. There are still many issues to explore. For example, we observed that extra bit rate savings by sub-pixel MC were rather moderate throughout the all the four sequences we tested. However, it is too early to draw any definitive conclusion about the effectiveness of sub-pixel accurate MC on lossless coding. The bottom line is that the more effective the sub-pixel accurate motion compensation is for reducing residual energy, the less bit rate would be needed for lossless compression. Other issues for future works include scaling for improved lossy regime PSNR performance and how to adjust the Lagrange multiplier used for RD optimal motion estimation mode decision in the MC-EZBC as it was found to strongly affect lossless compression ratios.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the National Geospatial-Intelligence Agency under Grant No. NMA201-00-1-2008.

REFERENCES

1. Sehoon Yea and William A. Pearlman,"A Wavelet-Based Two-Stage Near-Lossless Coder," *Proc. of IEEE Intl. Conf. on Image Processing*, Singapore, Oct., 2004.
2. Sehoon Yea, Sungadae Cho and William A. Pearlman,"Integrated lossy, near-lossless, and lossless compression of medical volumetric data ," *Proc. of SPIE Electronic Imaging 2005* , Vol. 5685, pp 151-159, San Jose, CA, Jan., 2005.
3. R. Ansari , N.Memon and E.Ceran,"Near-lossless image compression techniques," *Journal of Electronic Imaging*, Vol.7, No.3, pp 486-494, July, 1998.
4. Detlev Marpe and Gabi Blattermann and Jens Ricke and Peter Maab, "A two-layered wavelet-based algorithm for efficient lossless and lossy image compression," *IEEE Transactions on Circuits Syst. Video Technol.*, Vol. 10, No. 7, pp. 1094-1102, Oct. 2000.
5. T. Berger, *Rate-Distortion Theory*, Prentice-Hall, 1971.
6. Jianping Pan and Thomas R. Fischer, "Two-Stage Vector Quantization - Lattice Vector Quantization," *IEEE Trans. on Inform. Theory*, Vol. 41, No. 1, pp-155-163, 1995.
7. Padmanabha Rao and William A. Pearlman, "On Entropy of Pyramid Structures," *IEEE Trans. on Inform. Theory*, Vol. 37, No. 2, pp. 407-413, March 1991.
8. D. Brunello, G. Calvagno, G.A. Mian and R. Rinaldo,"Lossless Compression of Video Using Temporal Information," *IEEE Trans. on Image Processing*, Volume: 12 , Issue: 2, pp 132 - 139, Feb. 2003.
9. S. Andriani *et al.* ,"Comparison of Lossy to Lossless Compression Techniques for Digital Cinema," *Proc. of IEEE Intl. Conf. on Image Processing*, Singapore, Oct., 2004.
10. S. Park and Edward Delp ,"Adaptive Lossless Video Compression Using an Integer Wavelet Transform," *Proc. of IEEE Intl. Conf. on Image Processing*, Singapore, Oct., 2004.
11. S. Sun and S. Lei,"On Study of a Lossless Video Coding Algorithm Based on H.26L Tools ," *Proc. of SPIE Electronic Imaging 2003* , Vol. 5022, pp 994-1000, May, 2003.
12. Seishi Takamura and Yoshiyuki Yashima,"H.264-Based Lossless Video Coding Using Adaptive Transforms," *Proc. of ICASSP 2005*, Philadelphia, March, 2005.
13. Seishi Takamura and Yoshiyuki Yashima,"Lossless Scalable Video Coding with H.264 Compliant Base Layer," *Proc. of IEEE Intl. Conf. on Image Processing*, Genova, Italy, Sept., 2005.
14. "Requirements and Applications for Scalable Video Coding v.5," *MPEG/N6505*, Redmond, July, 2004.
15. "Scalable Video Coding Requirements for Video Surveillance Systems," *MPEG/M10822*, Redmond, July, 2004.
16. Ralf Schafer, Heiko Schwarz, Detlev Marpe, Thomas Schierl and Thomas Wiegand,"MCTF and Scalability Extension of H.264/AVC and its Application to Video Transmission, Storage, and Surveillance," To appear in *Proc. of Visual Communications and Image Processing (VCIP) 2005*, Beijing, China, July, 2005.
17. Mathias Wien, Thomas Rusert and Konstantin Hanke, "RWTH Proposal for Scalable Video Coding Technology," *MPEG/M10569/S16*, Munich, March, 2004
18. B. Pesquet-Popescu and V. Bottreau,"Three-dimensional lifting schemes for motion compensated video compression," Proc. IEEE ICASSP, May, 2001
19. P. Chen and J. W. Woods,"Bi-directional MC-EZBC with Lifting Implementation," *IEEE Trans. on Circ. and Sys. for Video Technology*, Volume: 14 , pp 1183 - 1194, October 2004.

20. B.J. Kim, Z. Xiong and W.A. Pearlman,“Low Bit-Rate Scalable Video Coding with 3-D SPIHT,”*IEEE Trans. on Circuits Syst. Video Technol.*,Vol.10, , pp.1374-1387,December 2000.
21. A. Said and W. Pearlman,“An Image Multiresolution Representation for Lossless and Lossy Compression,” *IEEE Trans. on Image Processing*, Vol. 5, pp. 1303-1310, Sept. 1996.
22. Zixiang Xiong, Xiaolin Wu, Samuel Cheng and Jianping Hua, “Lossy-to-Lossless Compression of Medical Volumetric Data Using Three-Dimensional Integer Wavelet Transforms,”*IEEE Trans. on Medical Imaging*, Vol. 22, No. 3, pp. 459-470, March 2003.
23. D. Taubman, “JPEG2000 : Image Compression Fundamentals, Standards and Practice,”,Massachusetts, Kluwer,2002.
24. Andrew Secker and David Taubman, “Highly Scalable Video Compression with Scalable Motion Coding,” *IEEE Trans. on Image Processing*, Vol. 13, No. 8, pp. 1029-1041, August 2004.
25. Kakadu Software : A Comprehensive Framework for JPEG 2000 (http://www.kakadusoftware.com)
26. I. H. Witten, R. M. Neal, J. G. Cleary, “Arithmetic Coding for Data Compression,” *Communication of ACM*, vol. 30, pp.520-540, June 1987.