ADAPTIVE LOSSLESS VIDEO COMPRESSION USING AN INTEGER WAVELET TRANSFORM

Sahng-Gyu Park and Edward J. Delp *

Video and Image Processing Laboratory
School of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana, USA

Haoping Yu

Corporate Research Group
Thomson, Indianapolis, Indiana, USA

ABSTRACT

In this paper, we describe an adaptive lossless compression algorithm for color video sequences utilizing backward adaptive temporal prediction and an integer wavelet transform. We exploit two redundancies in color video sequences, specifically spatial and temporal redundancies. We show that an adaptive scheme exploiting the two redundancies has better compression performance than lossless compression of individual image frames. The result of the proposed scheme is compared to current video compression algorithms.

1. INTRODUCTION

High compression efficiency using lossy compression is achieved by sacrificing the quality of the original video sequence. In some applications, such as digital cinema, the preservation of the original sequence is more important than compression efficiency, in these applications lossless compression is used.

There are three types of redundancies in color video sequences: spatial, spectral and temporal redundancy. Spatial redundancy exists among neighboring pixels in a frame, which can be used to predict pixel values in a frame from nearby pixels. The lossless image compression algorithm CALIC [1] uses prediction by a nonlinear gradient. JPEG-LS [2] uses the MED (Median Edge Detector) prediction which can detect edges in neighboring pixels.

If a sequence is described by the RGB color space, there are redundancies among the three color components. This is referred to as spectral redundancy. Carotti [3] decorrelates color components using a differential coding scheme, where two of the colors are represented by the differences with the reference color. In the JPEG-2000 standard[4], a RCT (Reversible Color Transform) is used to convert from RGB to YCrCb color space.

In lossy compression, block based temporal prediction by motion compensation provides very high coding gain where the technique decorrelates the frames. The data rate is further reduced by the quantization of the residual error frames. Motion vectors, needed to reconstruct the frame, are transmitted as overhead data. In lossless compression, the same motion compensation technique can be used to predict pixel values. Several techniques utilizing temporal correlation in addition to spatial and spectral correlation have previously been described [5] [6] [7] [3] [8].

*This work was partially supported by a grant from the Indiana 21st Century Research and Technology Fund. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu

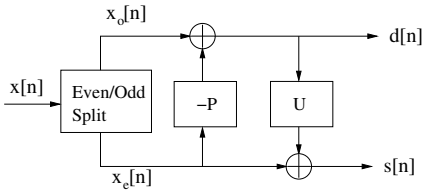

Fig. 1. Block Diagram of the Lifting Scheme: the Predict and Update Steps.

The wavelet transform has been used as a successful tool for image and video compression. In lossless compression, the complete recovery of the original pixel values is required. Thus an IWT (Integer Wavelet Transform), which maps integer pixels to integer coefficients, can be used [9] [10]. The IWT can recover the original values of the image without loss. The performance of the IWT highly depends on the content of the image and video and the wavelet filter used.

2. INTEGER WAVELET TRANSFORM

One approach used to construct the IWT (Integer Wavelet Transform) is the use of the lifting scheme (LS) described in [9]. The IWT construction using lifting is done in the spatial domain, contrary to the frequency domain implementation of a traditional wavelet transform[11].

The basic idea behind the lifting scheme is to exploit the correlation among the pixels. The correlation is typically local in space and frequency, i.e., adjacent pixels and spatial frequencies are more correlated than ones that are far apart[12]. A typical lifting scheme consists of the three steps shown in Figure 1: Split, Predict and Update. The lifting scheme is invertible, thus no information is lost. The reconstruction algorithm follows the same structure as the lifting scheme, but in reverse order.

The IWTs used in the experiments are[9]:

- A S transform:

$$\begin{aligned} d[n] &= x[2n+1] - x[2n] \\ s[n] &= x[2n] + \lfloor d[n]/2 \rfloor \end{aligned} \tag{1}$$

- A (2,2) transform:

$$\begin{aligned} d[n] &= x[2n+1] - \lfloor \tfrac{1}{2}(x[2n] + x[2n+2]) + \tfrac{1}{2} \rfloor \\ s[n] &= x[2n] + \lfloor \tfrac{1}{4}(d[n-1] + d[n]) + \tfrac{1}{2} \rfloor. \end{aligned} \tag{2}$$

- A (2+2,2) transform:

$$\begin{aligned} d^{(1)}[n] = &\quad x[2n+1] - \lfloor \tfrac{1}{2}(x[2n] + x[2n+2]) + \tfrac{1}{2} \rfloor \\ s[n] = &\quad x[2n] + \lfloor \tfrac{1}{4}(d^{(1)}[n-1] + d^{(1)}[n]) + \tfrac{1}{2} \rfloor \\ d[n] = &\quad d^{(1)}[n] - \lfloor \tfrac{1}{16}(-s[n-1] + s[n] \\ &\quad +s[n+1] - s[n+2]) + \tfrac{1}{2} \rfloor . \end{aligned} \tag{3}$$

- A (4,4) transform:

$$\begin{aligned} d[n] = &\quad x[2n+1] - \lfloor \tfrac{9}{16}(x[2n] + x[2n+2]) \\ &\quad -\tfrac{1}{16}(x[2n-2] + x[2n+4]) + \tfrac{1}{2} \rfloor \\ s[n] = &\quad x[2n] + \lfloor \tfrac{9}{32}(d[n-1] + d[n]) \\ &\quad -\tfrac{1}{32}(d[n-2] + d[n+1]) + \tfrac{1}{2} \rfloor \end{aligned} \tag{4}$$

3. NEW ADAPTIVE LOSSLESS VIDEO COMPRESSION ALGORITHM

Figure 2 shows the block diagram of the new proposed video encoder. A complete description of this algorithm is presented in [6]. If the video sequence is in the RGB color space, every frame of the color video sequence undergoes a RCT (reversible color transform) to exploit spectral redundancy. In the "Integer Wavelet Transform Selector" block, the most effective IWT (integer wavelet transform), in terms of lowest entropy, for each color component is determined using the 4 transforms described in the previous Section. One level of the selected IWT is obtained on the corresponding color component. The 3 selected IWT IDs, one for each color component, for a frame are transmitted to the decoder. Each subband is partitioned into $N \times N$ blocks. The MED (Median Edge Detector) spatial prediction[2] and *backward adaptive* temporal prediction[5] [6] are performed on each block. A frame buffer is used to store the previous frame. $e1$ and $e2$ are the residuals from the spatial prediction and the backward adaptive temporal prediction, respectively. In the "Block-Based Adaptive Selector" block, the prediction mode between spatial prediction and temporal prediction for each block is selected using the SAD (Sum of Absolute Difference). This is known as "adaptive selection of spatial and temporal prediction" and is described in the next section. The block prediction mode is then transmitted to the decoder. No side information, for example motion vectors, is transmitted for the temporal prediction.

In the "Subband Prediction Mode Selector" block, the subband prediction mode is determined using the data rate of the residuals obtained by the adaptive selection of spatial and temporal prediction and the data rate of the wavelet coefficients. The subband prediction mode is also transmitted to the decoder. If the data rate of the wavelet coefficients is smaller than that of the residual error, we label this as the *Direct Sending* mode. In the *Direct Sending* mode, no information concerning the block prediction mode and the residual error is transmitted. Instead, the wavelet coefficients of the subband are encoded. The other case is known as *Prediction* mode. In the *Prediction* mode, the residual error and the prediction mode for each block are transmitted. The above encoding procedure is repeated without using the IWT. In this case, "Subband" is replace by "Frame." In the "Adap-IWT Selector" block, the best approach between using IWT or not using IWT is determined. The "Context-Based Arithmetic Coder" block is used to encode the residual error or the wavelet coefficients according to the subband prediction mode.

At the decoder, the same procedure is performed in the reverse order to recover the original pixels. The decoding process can be done much fast than the encoding process since each operation mode is included in the bitstream.

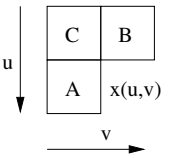

Fig. 3. The Neighbor Pixels of the Current Pixel $x(u, v)$ Used for Spatial Prediction.

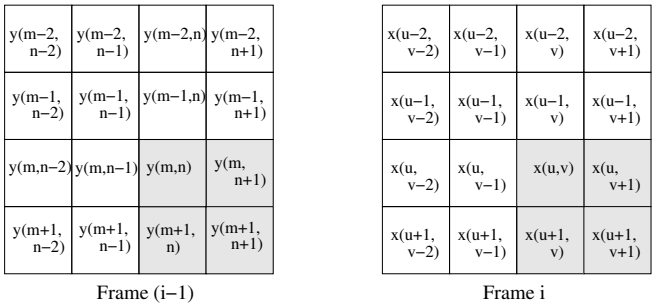

Fig. 4. The Neighbor Pixels of the Current Pixel $x(u, v)$ Used for Temporal Prediction.

3.1. Spatial Decorrelation

In order to remove spatial redundancy in a frame, predictive coding (DPCM) is used whereby an estimate of a pixel is obtained and the estimation error is then encoded. We use the MED(Median Edge Detector) predictor as described by JPEG-LS [2]. MED can detect horizontal and vertical edges by examining the left (A), the upper (B) and the upper-left (C) neighbors of the current pixel $x(u, v)$ as shown in Figure 3. This approach has been shown to be very robust across many different types of images. The prediction is the following:

$$\hat{x}(u,v) = \begin{cases} min(A,B) & \text{if } C \geq max(A,B) \\ max(A,B) & \text{if } C \leq min(A,B) \\ A+B-C & \text{otherwise.} \end{cases} \tag{5}$$

where $\hat{x}(u, v)$ is the prediction of $x(u, v)$. The MED predictor always selects either the best or the second-best predictor among the three candidate predictors.

3.2. Temporal Decorrelation

Temporal prediction removes redundancies between frames. In lossy compression, block based temporal prediction by motion compensation provides very high coding gain. Motion vectors, needed to reconstruct the frame, are transmitted as overhead data. Motion information is used in a slightly different manner in our method. To predict the motion from Frame $i-1$ to Frame i we divide the subband to be predicted into non-overlapping 2×2 blocks. In Figure 4 the shaded 2×2 block in Frame i is the block needed to be encoded using motion prediction based on knowing the previous Frame $i-1$ and other blocks in Frame i. This block will be

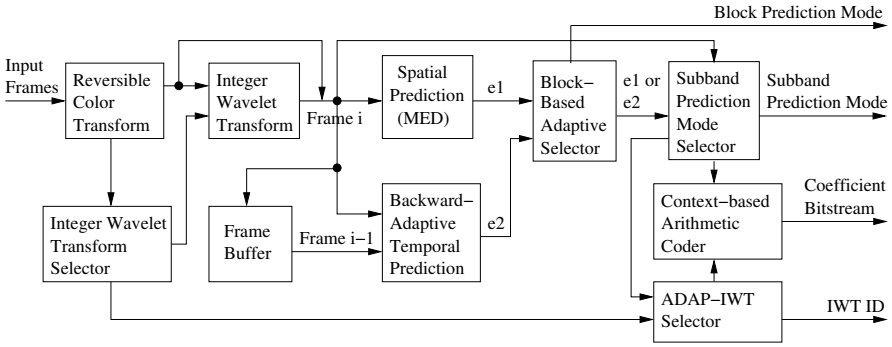

Fig. 2. Block Diagram of the Proposed Lossless Video Encoder.

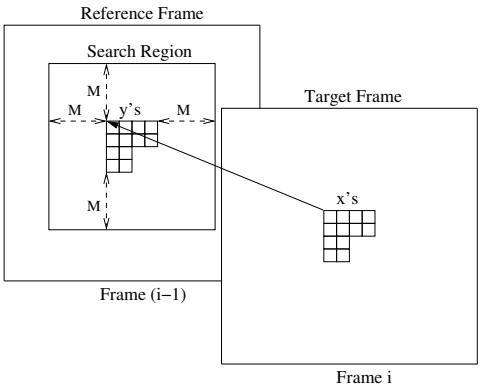

Fig. 5. Block Search for Temporal Prediction with Search Range M.

known as the "current block." The region used for the prediction consists of all the pixels except the shaded pixels of Frame i shown in Figure 4. This is denoted as the "target window." Thus the target window is a causal neighborhood of the current block. The pixels of the target window are known to both the encoder and decoder.

The motion search is performed as shown in Figure 5. The goal is to match the current target window from Frame i with a similarly defined target window in Frame $i-1$. For the work reported here; the search region for the best match is regions defined by the search ranges 0, 2, 4 and 6 pixels that bound the target window in Frame $i-1$. The SAD (Sum of Absolute Difference) is obtained between the current target window from Frame i and the reference target window from Frame $i-1$ for all the blocks in the search region as shown in Figure 5. Specifically for the current block $\{x(u+s, v+t)|s,t \in [0,1]\}$, the SAD is defined by the following:

$$SAD_{uv}(m,n) = \sum_{i=-1}^{-2}\sum_{j=-2}^{1} |x(u+i,v+j) - y(m+i,n+j)| + \sum_{i=0}^{1}\sum_{j=-2}^{-1} |x(u+i,v+j) - y(m+i,n+j)|, \quad (6)$$

where m ∈ [u-M, u+M], n ∈ [v-M, v+M] and M is the search range.

Indices k and l with the minimum SAD are obtained by the following:

$$k = argmin_m SAD_{uv}(m,n) \quad (7)$$
$$l = argmin_n SAD_{uv}(m,n).$$

The prediction, $\hat{x}$, of the current block is then determined by the target window with the smallest error. Therefore the predicted block for the current block is the following:

$$\hat{x}(u+s, v+t) = y(k+s, l+t), \qquad s,t \in [0,1] \quad (8)$$

The basic assumption is that if blocks in the target window have similar values between Frame i and Frame $i-1$ then their corresponding current block also has similar values as shown in Figure 4. If the SAD of non-shaded pixels between Frame i and Frame $i-1$ is minimum in the search area, then the shaded pixels in Frame i may be well predicted by the shaded pixels in Frame $i-1$.

The residual error is obtained by the following:

$$e(u+s, v+t) = x(u+s, v+t) - \hat{x}(u+s, v+t), \quad (9)$$

where s,$t \in$ [0,1]. This procedure is repeated for all blocks. The above procedure can be applied at the decoder side in the exact same way without any side information for the temporal prediction. Thus, it is known as *backward-adaptive* temporal prediction.

4. ADAPTIVE PREDICTION MODE SELECTION

If there is a great deal of motion in the video sequence, temporal prediction does not perform well with respect to compression efficiency (data rate). In this case, only spatial prediction is used. In other cases, temporal prediction may work better than spatial prediction. The decision whether spatial or temporal prediction is used can be made on a block-by-block basis. For a given block, each prediction technique is used (spatial and temporal) and the SAD (Sum of Absolute Difference) between the current and the predicted block is evaluated. The prediction technique with the smaller SAD is selected as the prediction technique for that block and the prediction mode is transmitted to the decoder as side information. One bit for each block is needed for the "prediction-type" flag. The overhead of sending the side information should be small. If the block size is 16×16 the entropy reduction by adaptive selection of the prediction mode should compensate for the overhead.

If an IWT is used, the above adaptive prediction mode is done on each subband in the wavelet domain. In this case, another adaptation step is needed to decide whether the original wavelet coefficients or the prediction residuals are encoded for each subband. If the data rate of the original wavelet coefficients is smaller than that of the prediction residuals obtained by the adaptive prediction mode, the wavelet coefficients are encoded and transmitted to the decoder. This is known as the *Direct Sending* mode. The prediction residuals and side information are encoded and transmitted, otherwise. This is known as the *Prediction* mode. Typically the LL subband uses the *Prediction* mode and the HL, LH and HH subbands use the *Direct Sending* mode. This information is also transmitted to the decoder.

5. EXPERIMENTAL RESULTS

Our lossless video encoder was tested on 13 color video sequences in the YUV color space. As a performance measure of the proposed video encoder, the data rates of the compressed video sequences after using an arithmetic encoder are obtained. In order to compare the performance of the proposed encoder, the test sequences are also encoded with the JPEG-LS [2] and the CALIC [1] lossless image compression techniques.

In Table 1, *JPEG-LS* and *CALIC* represent the data rate of the JPEG-LS and CALIC compression techniques respectively. *ADAP* represents the data rate of the compressed video sequence using the proposed algorithm. In the *ADAP* case, temporal prediction is used for macro blocks only if temporal prediction performed better than spatial prediction.

Table 1 shows the compression results for the YUV sequences with the various frame sizes. CALIC technique generally shows better result than JPEG-LS technique. The proposed algorithm shows excellent performance for QCIF video sequences. The average data rate reduction from *CALIC* to *ADAP* is 2.76 b/p, 0.65 b/p, 1.13 b/p and 1.12 b/p, which are 64.1.0 %, 13.2 %, 30.8 % and 18.7 % reductions for the "Akiyo," "Carphone," "Claire" and "Coastguard" sequences respectively. This reduction comes from the high temporal correlation between successive frames due to small amounts of motion in these sequences.

The"Fountain" sequence is a computer-generated sequence and does not have many changes in consecutive frames. Thus, its compression performance by *ADAP* is very high. The "Hockey" sequence contains very high motion and hence temporal correlation between consecutive frames is low. In this case, the contribution of temporal prediction is not large. Instead, the use of better spatial prediction techniques contributes more to the compression performance. Thus, *CALIC* shows better compression performance for this sequence since *CALIC* exploits spatial prediction more efficiently than *JPEG-LS*. For the other sequences, *ADAP* shows the best compression performance.

6. CONCLUSION

We presented a new algorithm for lossless video compression. The new algorithm incorporates an integer wavelet transform and removes temporal redundancy by the backward adaptive temporal prediction. Adaptive selection of prediction mode between spatial and temporal prediction is used to further improve the compression performance. The compression performance of the new algorithm is better than that of JPEG-LS and CALIC.

Table 1. Data Rates (Bits/pixel) for the YUV Sequences.

Sequence		JPEG-LS	CALIC	ADAP
176×144	Akiyo	4.330	4.310	**1.547**
	Carphone	4.975	4.934	**4.283**
	Claire	3.589	3.665	**2.533**
	Coastguard	6.018	5.989	**4.866**
352×288	Hall Monitor	4.904	4.814	**4.661**
	Paris	5.894	5.886	**3.567**
352×240	News2cut	4.195	4.161	**3.891**
	Tennis	7.305	7.145	**6.253**
720×480	Flowergarden	7.128	6.930	**6.403**
	Football	6.511	6.298	**6.224**
	Fountain	3.574	3.501	**1.136**
	Hockey	3.822	**3.642**	3.748
	Mobile	7.499	7.366	**6.708**

7. REFERENCES

[1] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Transactions on Communication*, vol. 45, no. 4, pp. 437–444, April 1997.

[2] JPEG-LS, *information Technology - Lossless and Near-lossless Compression of Continuous-tone Still Images*, 1998, Final Draft International Standard FDIS14495-1.

[3] E. S. G. Carotti, J. C. D. Martin, and A. R. Meo, "Backward-adaptive lossless compression of video sequences," *Proceedings of the IEEE International Conference on Audio, Speech and Sinal Processing*, vol. 4, pp. 3417–3420, May 2002.

[4] ITU-T, *ITU-T Recommendation T.800: CD15444-1, V1.0*, December 2000, (JPEG2000).

[5] Sahng Gyu Park and Edward J. Delp, "Adaptive lossless video compression," *Proceedings of SPIE Visual Communication and Image Processing*, pp. 246–254, January 2004.

[6] Sahng-Gyu Park, *Adaptive Lossless Video Compression*, Ph.D. thesis, Purdue University, School of Electrical and Computer Engineering, December 2003.

[7] A. J. Penrose and N. A. Dodgson, "Extending lossless image compression," *Eurographics UK '99*, April 1999.

[8] N. D. Memon and K. Sayood, "Lossless compression of video sequences," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1340–1345, Oct 1996.

[9] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.

[10] A. R. Calderbank, I. Daubechies, W. Sweldens, and B. Yeo, "Lossless image compression using integer to integer wavelet transforms," *IEEE International Conference on Image Processing*, vol. 1, pp. 596–599, October 1997.

[11] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[12] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.