Optimum Update for Motion-Compensated Lifting

Bernd Girod, *Fellow, IEEE,* Sangeun Han*, *Student Member, IEEE*

***Abstract*— In wavelet video coding with motion-compensated lifting, the update step typically reverses the motion vectors from the prediction step. Where motion compensation is not invertible, heuristic rules are used for the update step. This paper derives a closed-form expression for the optimal update step for a given general linear prediction step and applies the result to motion-compensated wavelet coding. Our analysis justifies using reversed motion vectors, where possible, and yields new results on proper inversion of subpixel motion compensation and optimal treatment of motion discontinuities. A new sparse matrix technique is presented that allows the practical implementation of the optimal update step for motion-compensated wavelet video coding. Experimental results confirm that optimizing the update step improves the rate-distortion performance for a motion-compensated wavelet video coder and provide justification for heuristics used in conventional techniques.**

I. Introduction

Three-dimensional subband coding (3D-SBC) of video sequences using wavelet transforms recently received much attention as an alternative to motion-compensated predictive coding. A key advantage is that 3D-SBC provides superior support for embedded, rate-scalable signal representations, often a requirement in best-effort networks. In an early attempt to incorporate motion compensation into 3D-SBC, Ohm [1] treats disconnected pixels differently from connected pixels to maintain invertibility in integer-pel accurate motion compensation. In this work, the temporal filter is limited to a two-tap Haar wavelet, which is unsatisfactory. Taubman and Zakhor [2] spatially align video frames by arbitrary frame warping before applying 3D-SBC. In general, this warping is not invertible, and therefore cannot achieve perfect reconstruction.

The technique of *motion-compensated lifting* incorporates unrestricted motion compensation into 3D-SBC in a reversible fashion [3] [4]. Lifting is a procedure to implement discrete wavelet transforms [5]. Input samples are first split into odd and even samples. In a *prediction step*, the odd samples are predicted from the even samples, resulting in a high-band signal. In an *update step*, the high-band signal is filtered and added back to the even sample sequence, resulting in a low-band signal. Because the lifting decomposition is easily invertible, any type of operation, linear or non-linear, can be incorporated into the prediction and update steps. The lifting implementation of the wavelet transform allows for a motion-compensated temporal transform, based on any wavelet kernel and any motion model, without sacrificing the perfect reconstruction property.

Motion compensation in the prediction step should minimize the bit-rate required to encode the temporal high-band pixel. Since this bit-rate is monotonically related to the energy of the high-band signal, this energy can be used instead to find the best motion vectors. This is analogous to minimizing residual error energy in motion-compensated predictive coding. The appropriate motion compensation for the update step, however, is not obvious. Various methods have been compared to compute backward motion fields for the update step [6], and some authors (e.g., [7]) have even reported that the update step degrades rate-distortion performance and should be omitted altogether, leading to a “truncated wavelet transform.” Typically, however, reversed motion vectors are used, in combination with heuristics for “unconnected” and “multiply connected” pixels [8] [9]. In [10], appearing also in this issue, the authors provide a theoretical analysis for optimizing the update step with integer-pel accurate motion compensation. They propose update steps that average multiply connected pixels and also discuss non-linear update steps. Their work is closely related to our paper.

This paper derives a closed-form expression for the optimal update step for a given general linear prediction step and applies the result to motion-compensated wavelet coding. Our analysis justifies using reversed motion vectors, where possible, and yields new results on proper inversion of subpixel motion compensation and optimal treatment of motion discontinuities. Section II presents our theoretical analysis of the optimal update step for a given motion-compensated prediction step. Section III provides examples describing various cases of motion vector fields. Section IV presents rate-distortion comparisons for encoding video sequences with a motion-compensated wavelet coder.

II. Optimum Prediction and Update Steps

We consider the decomposition of a video sequence into a temporal low-band and a temporal high-band signal, using a lifting implementation as shown in Fig. 1. The signals X and Y are the odd frames and the even frames of a video sequence, respectively. When considering a Haar wavelet, X and Y can also just be two frames, since the Haar basis vectors do not overlap. For our analysis, it is convenient to define X and Y as column vectors that contain all of the N pixel values in the respective odd-frame and even-frame-sequences,

EDICS: 2.IMMD

This work was supported, in part, by the Stanford Center for Integrated Systems and by the National Science Foundation under Grant No. ECS-0225315.

The authors are with the Information Systems Laboratory, Stanford University, Stanford, CA, USA. (email: bgirod@Stanford.EDU; sehan@Stanford.EDU)

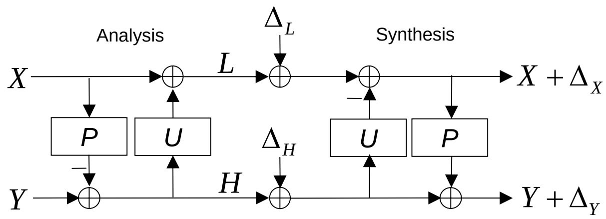

Fig. 1. Low-band/high-band decomposition by lifting.

for example, frame by frame in line-scan order. We assume that both prediction and update steps involve exclusively linear operations, hence they can be represented by premultiplying the signal vector by a matrix. The temporal high-band signal vector H is produced by $H = Y - PX$ where P is the $N \times N$ prediction matrix. The update step $L = X + UH$ multiplies H with the $N \times N$ update matrix U to yield the low-band signal L. Note that we omit possible scaling factors for the high-band and the low-band signals. These are not needed to derive the optimum update step in the following.

While the prediction step P can represent any linear operation, we think of it as representing motion-compensated prediction that strives to minimize the bit-rate required to encode H along with the motion vectors used for prediction. The bit-rate required to encode low-band signal L is typically the same as would be needed to encode X directly. Since the variance of H is so much smaller than the variance of X, the exact choice of the update step U has usually only a very small impact on the bit-rate required for L. When L is further decomposed in successive temporal transform stages, the frames are further apart and so the variance of H could increase. However, it is still relatively much lower than that of L, and therefore the influence of U on the bit-rate required for L is neglected in the following.

If U does not impact the bit-rate, why then use an update step at all? Should we omit the update step entirely and use the “truncated wavelet” as suggested in [7]? Inspection of the inverse transform (Fig. 1) reveals that U greatly impacts the distortion in the reconstructed even and odd frame sequences $X + \Delta_X$ and $Y + \Delta_Y$. We denote by Δ_L and Δ_H the quantization errors introduced by lossy source coding, and by

$$\begin{pmatrix} \Delta_X \\ \Delta_Y \end{pmatrix} = \begin{pmatrix} I \\ P \end{pmatrix} \Delta_L + \begin{pmatrix} -U \\ I - PU \end{pmatrix} \Delta_H \tag{1}$$

the resulting errors in the reconstructed frames.

We desire to choose U such that it minimizes the mean-squared error

$$D = E\{\Delta_X^T \Delta_X + \Delta_Y^T \Delta_Y\} \tag{2}$$

We may assume that Δ_L and Δ_H are uncorrelated random vectors. Then, the choice of U does not affect the part of the distortion D due to Δ_L and we only have to consider the contribution of the high-band error. Without the loss of generality, Δ_L is set to zero, i.e.,

$$\begin{pmatrix} \Delta_X \\ \Delta_Y \end{pmatrix} = \begin{pmatrix} -U \\ I - PU \end{pmatrix} \Delta_H \tag{3}$$

Consider the autocorrelation matrix

$$\begin{aligned} R &= E\left\{ \begin{pmatrix} \Delta_X \\ \Delta_Y \end{pmatrix} \begin{pmatrix} \Delta_X \\ \Delta_Y \end{pmatrix}^T \right\} \\ &= \begin{pmatrix} -U \\ I - PU \end{pmatrix} E\left\{\Delta_H \Delta_H^T\right\} \begin{pmatrix} -U \\ I - PU \end{pmatrix}^T . \end{aligned} \tag{4}$$

With the autocorrelation matrix of Δ_H, $R_H = E\{\Delta_H \Delta_H^T\}$, we write the expression as

$$R = \begin{pmatrix} -U \\ I - PU \end{pmatrix} R_H \begin{pmatrix} -U \\ I - PU \end{pmatrix}^T \tag{5}$$

We find the optimum U by matrix calculus

$$\begin{aligned} d/dU\,(D) &= d/dU\,(tr\,(R)) \\ &= 2\left(I + P^T P\right) U R_H - 2P^T R_H = 0 \end{aligned} \tag{6}$$

where $d/dU(.)$ is a matrix whose (i,j) element is the derivative of the argument with respect to the (i,j) element of U. Assuming that R_H is full rank, we find the update matrix

$$U = (I + P^T P)^{-1} P^T \tag{7}$$

corresponding to a local extremum of the mean-squared error D (2). By perturbing the solution and plugging $(I + P^T P)^{-1} P^T + \Delta U$ into (5), the reader might verify that (7) represents the global minimum. Further, we note that $(I + P^T P)$ is positive definite and therefore always invertible.

Once we find the prediction step P that minimizes the bit-rate, we can easily determine the corresponding update step that minimizes the resulting distortion using (7). Except that we require the prediction step to be linear, there are no constraints on P. Hence, (7) can tell us how to best handle sub-pixel interpolation, non-invertible motion vector fields with unconnected or multiply-connected pixels, overlapped block motion compensation, and any combination of these linear techniques. In the next section, we study some of these cases in more detail.

III. Optimal Update Examples

While (7) can be applied to a lifting decomposition with any wavelet kernel, we restrict ourselves to a temporal Haar transform in the following examples. The vectors X and Y are then simply two successive frames. After motion-compensated lifting, the high-band signal H corresponds to frame Y, and the low-band signal L corresponds to X. With block-wise motion compensation, the frame Y is divided into blocks using a fixed grid, and each block finds a best match in frame X. If only integer displacements are allowed, most pixels in frame X are connected to one pixel in Y (*1-connected*). However, due to the spatial variation of the displacement vector, some pixels in frame X may be used for prediction more than once (*multi-connected, or M-connected pixels*), while others may not be used at all (*unconnected pixels*).

Fig. 2 illustrates the optimal update step for a one-dimensional example of integer-pixel-accurate motion compensation. The circles are pixels and the arrows are the direction of motion compensation in the lifting steps. In Fig. 2, (a) is the prediction step while (b) is the optimal update step

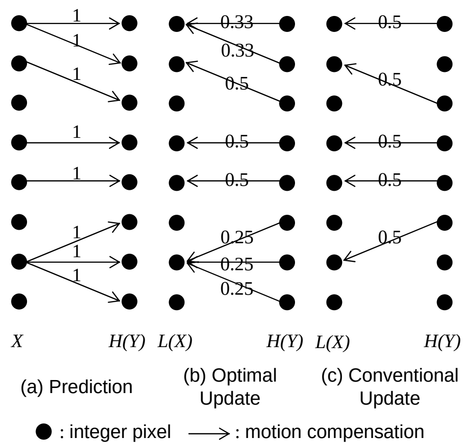

Fig. 2. Example: optimal update for integer-pel accuracy motion compensation. $L(X)$ and $H(Y)$ denote references for temporal low/high-bands.

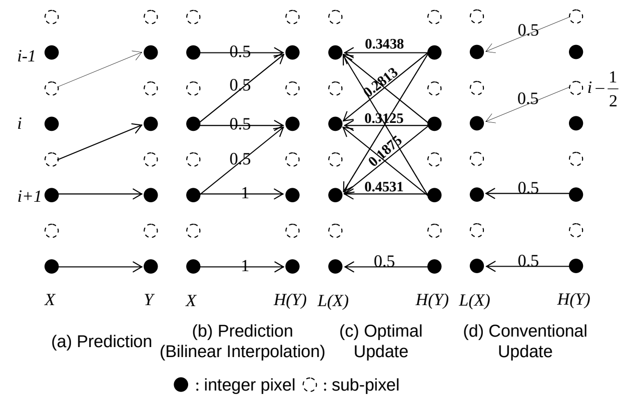

Fig. 3. Example: optimal update for half-pel accuracy motion compensation. $L(X)$ and $H(Y)$ denote temporal references for temporal low/high-bands. (In (c), only weights with absolute values > 0.1 are shown.)

based on the (7) in Section II. The optimal solution illustrates three simple rules which are easily derived from (7):

- If a pixel in X is 1-connected, the corresponding pixel in the high-band H is added to the pixel along the reversed motion vectors with a weight of $\frac{1}{2}$.
- If a pixel in X is unconnected, this pixel is simply copied to the corresponding position in L.
- If a pixel in X is M-connected, all the connected pixels in the high-band H are added to the pixel with a weight of $\frac{1}{M+1}$ [1]. 1-connected pixels are included as the special case $M = 1$.

Fig. 2(c) shows the best heuristic update known to us prior to this work [8]. We shall refer to it as the "conventional update step." For integer-pixel accuracy, 1-connected and unconnected pixels are treated in the same way as in the optimal case. However, M-connected pixels are treated in a different way. For instance, the first encountered pixel in Y that uses the current M-connected pixel in X as a predictor can be chosen for computing the low-band L. Another proposal found in the literature [9] is to choose the pixel in Y that best matches the reference pixel in X. This may cause a mismatch due to the quantization errors at the decoder. Once a pixel has been selected for the update step, the same weight of $\frac{1}{2}$ is used.

A one-dimensional example of half-pixel-accurate motion compensation is shown in Fig. 3. Bilinear interpolation for sub-pixel positions is used. The resulting weights needed for the matrix P are shown in (b). Pixels in both frames X and Y might be M-connected. However, there might still be 1-connected pixels and unconnected pixels, as in the integer-pixel-accuracy case, and their optimal update step is the same as in that case. However, M-connected pixels can result in many more pixels involved in generating the optimum low-band signal at these pixel locations. In optimum update step (c), only weights with absolute values greater than 0.1 are shown. Weights not shown are not necessarily zero.

In Fig. 3(d), the conventional update step is shown [9]. When the motion displacement points to a sub-pixel position in the frame X such as $X(i+\frac{1}{2})$, conventional techniques default to the nearest integer-pixel position $X(i)$, and use $H(i-\frac{1}{2})$ for the update step. Note that bilinear interpolation in the high-band yields similar weights to those in the optimal update.

Note that the proposed method can be applied to more general cases even though the Haar transform and the block-based motion compensation are used in the above examples. For longer filters such as 5/3 bi-orthogonal transform, U can be obtained from a finite number of neighboring Ps with a reasonable increase in computational complexity. Other motion models can be used as long as they are linear operators.

IV. Experimental Results

The experiments in this section compare the optimum update step given by (7) in Section II with the conventional update step [9]. Fig. 4 shows the encoder of the wavelet video coding system with motion-compensated lifting. We use variable-blocksize motion estimation [11] and furthermore, we employ fractional-pel accuracy motion compensation with bilinear interpolation. After temporal decomposition, to further exploit the coherence among neighboring pixels within each temporal subband, a multi-level 2-D spatial DWT is then applied to decompose the subband into wavelet coefficients. Finally, the SPIHT (Set Partitioning in Hierarchical Trees) [12] algorithm is used to encode the wavelet coefficients of each temporal subband into a scalable bitstream. The SPIHT algorithm provides an embedded representation so that rate-distortion curves can be obtained by simply truncating the coded bitstreams.

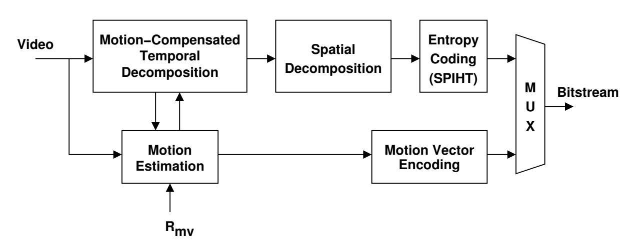

Fig. 4. Encoder structure of scalable interframe wavelet video coding system

For the optimal update scheme, the prediction matrix P can be populated from the block-wise motion vectors. Like

[1]The same result is derived in [10], however, unnecessarily assuming independent identically distributed (*i.i.d.*) quantization errors in the temporal high-band.

the encoder, the decoder only needs the motion vectors to construct P, so this matrix P need not be transmitted. P is a very large, but sparse matrix; e.g., 25344×25344 for QCIF sequences (176×144 pixels) in the case of temporal Haar transform. Obtaining the optimum update matrix U is challenging since the inverse of a sparse matrix is not necessarily sparse. In our implementation, we therefore do not calculate U explicitly. Instead, we observe that only UH is needed to produce the temporal low-band L. Our computation then proceeds as follows:

- Set $Z = UH = (I + P^T P)^{-1} P^T H$
- We note $(I + P^T P)Z = P^T H$.
- Solve $AZ = B$ where $A = I + P^T P$ and $B = P^T H$, exploiting the sparseness of P [13].
- Add Z to X.

Fig. 5 shows luminance PSNR over the total bitrate for two test sequences, *Foreman* and *Mobile & Calendar*, both consisting of 288 frames in QCIF format, encoded with three levels of temporal decomposition with the motion-compensated Haar transform. We use the same motion accuracy of integer, half and quarter-pel at all decomposition levels. We observe that the optimal update step performs only slightly better than the conventional update step. This justifies the heuristics used in the conventional update step [9].

V. Conclusion

We have derived a closed-form expression for the optimal update step for a given general linear prediction step and applied the result to motion-compensated wavelet coding. Our analysis provides justification for using reversed motion vectors, where possible. Unconnected and 1-connected pixels are treated as in the conventional update scheme. However, multiply-connected pixels can result in numerous pixels with different weights involved in generating the best low-band signal. A new sparse matrix technique is presented that allows the practical implementation of the optimal update step for motion-compensated wavelet video coding. Experimental results show that the optimal update outperforms the conventional update step by at most 0.4dB, and that conventional update using the reversed motion vectors does nearly as well as the optimal update, thereby justifying the heuristics used in the conventional method.

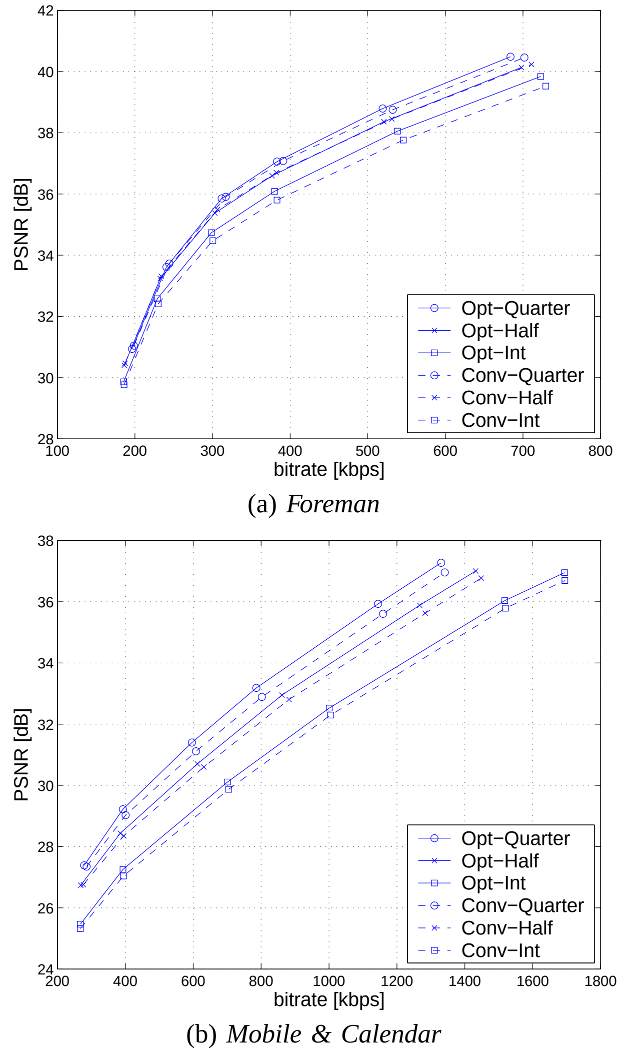

Fig. 5. Rate-Distortion performance with optimal and conventional update step. Integer, half, quarter-pel accurate motion compensations are used.

References

[1] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, Sept. 1994.

[2] D. Taubman and A. Zakhor, "Multirate 3-D subbnad coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, Sept. 1994.

[3] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001.

[4] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Proceeding (ICASSP)*, Salt Lake City, UT, May 2001.

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

[6] N. Božinović, J. Konrad, T. André, M. Antonini, and M. Barlaud, "Motion-compensated lifted wavelet video coding : Toward optimal motion/transform configuration," in *12th European Signal Processing Conference*, Vienna, Austria, Sept. 2004, To appear.

[7] L. Luo, S. Li, Z. Zhuang, and Y.-Q. Zhang, "Motion compensated lifting wavelet and its application in video coding," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, Oct. 2001.

[8] V. Bottreau, E. Barrau, and A. Bourge, "Architecture and features of a fully scalable motion-compensated 3d subband codec," ISO/IEC JTC1/SC29/WG11, MPEG2002/M7977, Jeju, Korea, March 2002.

[9] J. W. Woods and P. Chen, "Improved MC-EZBC with quarter-pixel motion vectors," ISO/IEC JTC1/SC29/WG11, MPEG2002/M8366, Fairfax, VA, May 2002.

[10] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, "Improved update operators for lifting-based motion-compensated temporal filtering," in *submitted to IEEE Signal Proc. Letters*, May 2004.

[11] ITU-T Rec. H.264 / ISO/IEC 11496-10, "Advanced Video Coding, Final Committee Draft," Document JVT-E022, Sept. 2002.

[12] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[13] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, Mar. 1982.