(N,0) MOTION-COMPENSATED LIFTING-BASED WAVELET TRANSFORM

Thomas André, Marco Cagnazzo, Marc Antonini, Michel Barlaud

Laboratoire I3S - UPRES-A 6070 CNRS
Université de Nice - Sophia Antipolis
2000 rte des Lucioles, F-06903 Sophia Antipolis, FRANCE
{andret,cagnazzo,am,barlaud}@i3s.unice.fr

Nikola Božinović, Janusz Konrad

Dept. of Electrical and Computer Engineering
Boston University
8 Saint Mary's St., Boston, MA 02215, USA
{nikolab,jkonrad}@bu.edu

ABSTRACT

Motion compensation has been widely used in both DCT- and wavelet-based video coders for years. The recent success of temporal wavelet transform based on motion-compensated lifting suggests that a high-performance, scalable wavelet video coder may soon outperform best DCT-based coders. As recently shown, however, the motion-compensated lifting does not implement exactly its transversal equivalent unless certain conditions on motion are satisfied. In this paper, we review those conditions, and we discuss their importance. We derive a new class of temporal transforms, the so-called 1-N transversal or (N,0) lifting transforms, that are particularly interesting if those conditions on motion are not satisfied. We compare experimentally the 1-3 and 5-3 motion-compensated wavelet transforms for the ubiquitous block-motion model used in all video compression standards. For this model, the 1-3 transform outperforms the 5-3 transform due to the need to transmit additional motion information in the later case. This interesting result, however, does not extend to motion models satisfying the transversal/lifting equivalence conditions.

1. INTRODUCTION

Wavelet-based video coders can be classified as 2-D subband coders, that apply only a spatial wavelet transform, and 3-D subband coders, that apply a spatio-temporal wavelet transform [1–6]. While the former category has not shown a great promise, the latter has been steadily improving and currently can achieve performance comparable to the best DCT-based coders. One particular improvement that recently raised expectations of 3-D wavelet coders is a combined use of lifting and motion compensation [7–9]. While the application of lifting steps instead of transversal implementation reduces the computational complexity, the inclusion of motion compensation facilitates temporal subband decomposition along motion trajectories [10]. This reduces the wavelet coefficient energy in both subbands thus leading to more efficient compression.

As recently shown [11], the use of motion compensation in lifting wavelet transform modifies its nature; a motion-compensated lifting wavelet transform is equivalent to motion-compensated transversal wavelet transform only if certain conditions on motion are satisfied. For example, for 2-2 (Haar) transform, this condition is motion invertibility, while for 5-3 transform in addition to invertibility motion additivity is also needed. An exact fulfillment of these conditions may not be possible for some motion models (e.g., block-based motion) and be computationally expensive for other models (e.g., deformable meshes). It is unclear what are the trade-offs in case motion invertibility/additivity cannot be achieved. If motion is constrained to be invertible/additive, it may not be very accurate and may result in less efficient motion compensation (poorer alignment of features). On the other hand, if the motion is not invertible but very accurate, the motion compensation error will be small, but since the lifting transform does not implement the corresponding transversal transform, subband decomposition may be inaccurate. It is unclear what happens between those two extremities, and how much an adjustment of motion transformation (accuracy, degree of departure from invertibility) may impact compression performance.

We explore some of the above issues in this paper. We review lifting/transversal equivalence conditions under motion compensation. We discuss relative merits of motion accuracy and invertibility. We derive a new class of temporal transforms, the so-called 1-N transversal or (N,0) lifting transforms, that are particularly interesting if motion invertibility/additivity does not hold. We compare experimentally the 1-3 and 5-3 motion-compensated wavelet transforms for the ubiquitous block-motion model used in all video compression standards. For this model, the 1-3 transform outperforms the 5-3 transform due to the need to transmit additional motion information in the later case. This interesting result, however, does not extend to motion models satisfying the transversal/lifting equivalence conditions.

2. MOTION COMPENSATION IN (2,2) LIFTING SCHEME

2.1. Lifting implementation

Lifting is an efficient way to implement wavelet transforms [12, 13]. For each transversal implementation of a wavelet filter, there exists an equivalent lifting-based implementation of the very same filter [14]. An interesting property of the lifting implementation is that it allows to introduce non-linear computations, such as motion compensation, and still assure perfect reconstruction. The inclusion of motion compensation in the lifting steps has been shown to improve efficiency of temporal subband decomposition [7, 8].

Let $(x_n)_K$ be an K-image sequence. Let's denote by $\mathbf{v}_{i+j\to i}(\mathbf{m})$ a motion vector of pixel at spatial location $\mathbf{m}$ in frame $i+j$ that displaces this pixel to new location $\mathbf{m}+\mathbf{v}_{i+j\to i}(\mathbf{m})$ in frame i. Estimation of this vector is usually based on the assumption of constant image intensity along motion trajectory, i.e., $x_i[\mathbf{m}+\mathbf{v}_{i+j\to i}(\mathbf{m})] \approx x_{i+j}[\mathbf{m}]$. The motion vector $\mathbf{v}_{i+j\to i}(\mathbf{m})$

is a forward (backward) vector if j is negative (positive). Thus, $x_i[\mathbf{m} + \mathbf{v}_{i+j\to i}(\mathbf{m})]$ is a motion-compensated image x_i with respect to image x_{i+j}.

In the following, we denote by (N, M) a lifting scheme where N and M are the lengths of the prediction and update operators, respectively [15]. Let's consider the (2,2) lifting transform. The lowpass and highpass sub-sequences, $(l_n)_{K/2}$ and $(h_n)_{K/2}$, respectively, are computed by means of the following steps:

$$\begin{aligned} \hat{h}_k[\mathbf{m}] &= x_{2k+1}[\mathbf{m}] - \frac{1}{2}\big(x_{2k}[\mathbf{m} + \mathbf{v}_{2k+1\to 2k}(\mathbf{m})] \\ &\quad + x_{2k+2}[\mathbf{m} + \mathbf{v}_{2k+1\to 2k+2}(\mathbf{m})]\big), \\ \hat{l}_k[\mathbf{m}] &= x_{2k}[\mathbf{m}] + \frac{1}{4}\big(\hat{h}_{k-1}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})] \\ &\quad + \hat{h}_k[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]\big). \end{aligned} \tag{1}$$

The original image sequence is recovered by motion-compensated synthesis lifting steps:

$$\begin{aligned} x_{2k}[\mathbf{m}] &= \hat{l}_k[\mathbf{m}] - \frac{1}{4}\big(\hat{h}_{k-1}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})] \\ &\quad + \hat{h}_k[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]\big), \\ x_{2k+1}[\mathbf{m}] &= \hat{h}_k[\mathbf{m}] + \frac{1}{2}\big(x_{2k}[\mathbf{m} + \mathbf{v}_{2k+1\to 2k}(\mathbf{m})] \\ &\quad + x_{2k+2}[\mathbf{m} + \mathbf{v}_{2k+1\to 2k+2}(\mathbf{m})]\big). \end{aligned} \tag{2}$$

Note that by design, perfect reconstruction is achieved regardless of properties of the motion field. However, it turns out that the motion-compensated lifting-based (2,2) wavelet transform described above does not implement transversal equations of the original wavelet unless the motion field $\mathbf{v}$ satisfies certain conditions [11].

2.2. Equivalence conditions

The transversal implementation of the motion-compensated 5-3 wavelet transform in case of the lowpass output is as follows:

$$\begin{aligned} l_k[\mathbf{m}] = \frac{3}{4}x_{2k}[\mathbf{m}] \quad &+ \quad \frac{1}{4}\big(x_{2k-1}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})] \\ &\qquad +x_{2k+1}[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]\big) \\ &- \quad \frac{1}{8}\big(x_{2k-2}[\mathbf{m} + \mathbf{v}_{2k\to 2k-2}(\mathbf{m})] \\ &\qquad +x_{2k+2}[\mathbf{m} + \mathbf{v}_{2k\to 2k+2}(\mathbf{m})]\big) \end{aligned}$$

whereas its lifting-based equivalent, expressed in terms of $(x_{2k})_{K/2}$ and $(x_{2k+1})_{K/2}$, is:

$$\begin{aligned} \hat{l}_k[\mathbf{m}] \quad = \quad & x_{2k}[\mathbf{m}] \\ -\tfrac{1}{8}\big(\quad & x_{2k}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m}) \\ & +\mathbf{v}_{2k-1\to 2k}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})]\big] \\ & +x_{2k}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m}) \\ & +\mathbf{v}_{2k+1\to 2k}[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]\big] \quad \big) \\ +\tfrac{1}{4}\big(\quad & x_{2k-1}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})\big] \\ & +x_{2k+1}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})\big] \quad \big) \\ -\tfrac{1}{8}\big(\quad & x_{2k-2}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m}) \\ & +\mathbf{v}_{2k-1\to 2k-2}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})]\big] \\ & +x_{2k+2}\big[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m}) \\ & +\mathbf{v}_{2k+1\to 2k+2}[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]\big] \quad \big) \end{aligned}$$

In order that transversal and lifting implementations produce the same result, one needs $\hat{l}_k[\mathbf{m}] = l_k[\mathbf{m}]$ (the highpass equations are identical). Considering that this must hold for *arbitrary* input sequence $(x_n)_K$, the following conditions must hold with respect to the motion field $\mathbf{v}$:

- Motion field $\mathbf{v}$ must be *invertible*:

  $$\begin{aligned} \mathbf{v}_{2k\to 2k-1}(\mathbf{m}) &= -\mathbf{v}_{2k-1\to 2k}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})], \\ \mathbf{v}_{2k\to 2k+1}(\mathbf{m}) &= -\mathbf{v}_{2k+1\to 2k}[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})]. \end{aligned}$$

  This problem has been acknowledged in the literature [10]. Invertibility condition is not satisfied, in general, by many popular motion models, such as those based on blocks. We discuss this further in the next section.

- Motion fields $\mathbf{v}$ must be *additive*:

  $$\begin{aligned} \mathbf{v}_{2k\to 2k-1}(\mathbf{m}) &+ \mathbf{v}_{2k-1\to 2k-2}[\mathbf{m} + \mathbf{v}_{2k\to 2k-1}(\mathbf{m})] \\ &= \mathbf{v}_{2k\to 2k-2}(\mathbf{m}), \\ \mathbf{v}_{2k\to 2k+1}(\mathbf{m}) &+ \mathbf{v}_{2k+1\to 2k+2}[\mathbf{m} + \mathbf{v}_{2k\to 2k+1}(\mathbf{m})] \\ &= \mathbf{v}_{2k\to 2k+2}(\mathbf{m}). \end{aligned}$$

  This condition states that motion between frames x_m and x_l, added to the motion between frames x_l and x_n, results in motion between frames x_m and x_n. However, as we shall see in the next section, this condition is not satisfied in many cases.

2.3. Equivalence violations

In general, invertible motion fields have to satisfy the following constraint: $\mathbf{v}_{i\to i+1}(\mathbf{m}) = -\mathbf{v}_{i+1\to i}(\mathbf{m} + \mathbf{v}_{i\to i+1}(\mathbf{m}))$. It is clear that not all motion fields are invertible. For example, motion fields estimated using popular block-based motion models (used in MPEG compression standards) are not invertible, particularly in unpredictable areas such as those due to occlusions, zooming, etc. This happens because the union of all translated blocks does not cover the complete image (see for [11] for more detailed discussion); there are gaps and overlaps between motion-compensated blocks. This problem has been recognized by Ohm [1] and others.

As for the additive motion fields, they must satisfy the following general relationship: $\mathbf{v}_{i\to i+1}(\mathbf{m}) + \mathbf{v}_{i+1\to i+2}\big(\mathbf{m} + \mathbf{v}_{i\to i+1}(\mathbf{m})\big) = \mathbf{v}_{i\to i+2}(\mathbf{m})$. However, this is rarely the case. Again, not all motion fields satisfy this condition, and block motion model is again an example. Since the union of all blocks translated from image x_m to image x_l does not cover the image completely (gaps exist), blocks in x_l that have a motion vector associated with them but which include these gaps have not corresponding vector in x_m to be added to. Clearly, additivity does not hold for block-based motion models. Other scenarios are possible as well.

If either invertibility or additivity is not satisfied, motion-compensated lifting steps (1-2) implement a wavelet transform that is different from the intended motion-compensated transversal implementation.

Although constraining motion vectors may allow them to satisfy invertibility or additivity or both, such a solution has its drawbacks. First, motion vectors that satisfy both conditions may be difficult to compute. Secondly, and more importantly, motion vectors computed under those constraints may not accurately model the underlying motion observed in the image sequence, and thus diminish the efficiency of motion compensation. This will lead to an increase in energy contained in the highpass temporal subband and reduce compression efficiency.

3. (N,0) TEMPORAL WAVELET TRANSFORM

3.1. General (N,0) filtering framework

As shown by Konrad [11], it is possible to design a lifting scheme that perfectly implements the motion-compensated transversal Haar transform without any conditions on motion transformations. Unfortunately, for longer filters such as the 5-3, the equivalence between lifting and transversal wavelet transforms depends on properties that should be, but often are not, satisfied by the motion fields.

It is impossible to build a transform that applies the original motion-compensated wavelet and scale function while assuring perfect reconstruction regardless of motion fields. Relaxing one of these constraints is necessary. For arbitrary motion, lifting-based transforms don't apply the original motion-compensated wavelet and transversal implementations are not invertible. Is there another alternative?

Let us consider the general case of motion-compensated two-step lifting scheme, represented by the following equations:

$$\hat{h}_k[\mathbf{m}] = x_{2k+1}[\mathbf{m}] + \sum_{i \in D_h} \alpha_i x_{2k+2i}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+2i}(\mathbf{m})],$$
$$\hat{l}_k[\mathbf{m}] = x_{2k}[\mathbf{m}] + \sum_{j \in D_l} \beta_j \hat{h}_{k+j}[\mathbf{m} + \mathbf{v}_{2k \to 2k+1+2j}(\mathbf{m})],$$

where D_h (respectively, D_l) is the support of the highpass (respectively, lowpass) filter.

Suppose we can estimate motion very accurately, and thus we have: $x_i[\mathbf{m} + \mathbf{v}_{i+j \to i}(\mathbf{m})] = x_{i+j}[\mathbf{m}] + \epsilon_{i+j}(i)[\mathbf{m}]$ with the error $\epsilon_{i+j}(i)[\mathbf{m}] \ll x_{i+j}[\mathbf{m}]$. The error term takes into account unpredictable parts of the image sequence, but we assume it still has a very low energy. Under this assumption, we can rewrite the previous equations as follows:

$$\hat{h}_k[\mathbf{m}] = x_{2k+1}[\mathbf{m}] + \sum_{i \in D_h} \alpha_i x_{2k+2i}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+2i}(\mathbf{m})]$$
$$\hat{l}_k[\mathbf{m}] = x_{2k}[\mathbf{m}] + \epsilon_{\hat{l}}(k)[\mathbf{m}],$$

where the error term $\epsilon_{\hat{l}}(k)[\mathbf{m}]$ is very small.

The highpass subband $\hat{h}_k[\mathbf{m}]$ contains errors due to imprecise motion compensation, while the lowpass subband contains the even-sampled subsequence plus another motion compensation error. Since $\epsilon_{i+j}(i)[\mathbf{m}] \ll x_{i+j}[\mathbf{m}]$, it is clear that $\epsilon_{\hat{l}}(k)[\mathbf{m}] \ll x_{2k}[\mathbf{m}]$ and $\hat{l}_k[\mathbf{m}] \approx x_{2k}[\mathbf{m}]$. Thus, $x_{2k}[\mathbf{m}]$ is a reasonable approximation of $\hat{l}_k[\mathbf{m}]$. We can, therefore, write:

$$\hat{h}_k[\mathbf{m}] = x_{2k+1}[\mathbf{m}] + \sum_{i \in D_h} \alpha_i x_{2k+2i}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+2i}(\mathbf{m})]$$
$$\hat{l}_k[\mathbf{m}] = x_{2k}[\mathbf{m}],$$

This "modified lifting" (N,0) transform is interesting since, regardless of the motion transformation, it is exactly equivalent to an 1-N' transversal wavelet transform with $N' = 2N - 1$. Moreover, it assures perfect reconstruction without any conditions on motion.

3.2. (2,0) lifting

For example, we derive the (2,0) transform from the original (2,2) lifting transform:

$$\begin{aligned}\hat{h}_k[\mathbf{m}] &= x_{2k+1}[\mathbf{m}] - \frac{1}{2}\big(x_{2k}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k}(\mathbf{m})] \\ &\quad + x_{2k+2}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+2}(\mathbf{m})]\big) \\ \hat{l}_k[\mathbf{m}] &= x_{2k}[\mathbf{m}]\end{aligned}$$

The above equations are also known as 1-3 lifting [10] and MC-LIFT (or truncated 5-3) [5]. The transform consists of the same highpass filter as in the regular 5-3 transform (the first equation in (1)), associated with a simple downsampling operation instead of lowpass filtering and downsampling. This approach has already been exploited as a temporal transform in fast coders, because for non-invertible motion only one motion field per frame needs to be computed for the 1-3 lifting transform in contrast to two motion fields in the case of the 5-3 lifting transform. A related benefit is that for the (2,0) transform only half of the motion vectors need to be encoded as compared to the usual 5-3 transform. These rate savings do not come free since the (2,0) transform has significant overlap of frequency subbands and thus $\hat{l}_k$ carries significant amount of information from $\hat{h}_k$ (this can be thought of as additional aliasing due to lack of lowpass filtering). If motion is inaccurately computed, however, the error $\epsilon_{\hat{l}}(k)[\mathbf{m}]$ may be quite large ("ghosting" in the lowpass subsequence), and, in fact, comparable to the additional aliasing error. This may be particularly true for longer wavelets that require motion compensation across many frames (5-3 wavelet requires motion compensation across 4 frames: x_{2k-1}, x_{2k}, x_{2k+1}, x_{2k+2}). Note, that the above considerations are not valid for invertible/additive motion models.

3.3. (4,0) lifting

From the general (N,0) framework, we derive also the (4,0) lifting-based transform adapted from the usual (4,2) lifting transform:

$$\begin{aligned}\hat{h}_k[\mathbf{m}] &= x_{2k+1}[\mathbf{m}] - \frac{9}{16}\big(x_{2k}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k}(\mathbf{m})] \\ &\quad + x_{2k+2}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+2}(\mathbf{m})]\big) \\ &\quad + \frac{1}{16}\big(x_{2k-2}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k-2}(\mathbf{m})] \\ &\quad + x_{2k+4}[\mathbf{m} + \mathbf{v}_{2k+1 \to 2k+4}(\mathbf{m})]\big) \\ \hat{l}_k[\mathbf{m}] &= x_{2k}[\mathbf{m}]\end{aligned}$$

Similarly to the 5-3 transform, in case of non-invertible/non-additive motion it would require two motion vector fields per frame.

4. EXPERIMENTAL RESULTS

In order to evaluate relative merits of the (2,0) and (2,2) temporal lifting, we have carried out experiments on the CIF sequence "Foreman" (30 fps). We compressed the frames of individual subbands using an intra-frame scan-based lifting encoder developed by us earlier [16]. We used 3 scan-based [13] temporal decomposition levels. The motion was computed using a standard 16×16 block matching, comparable to those used by MPEG and H.26* video coders, with full pixel precision. The resulting vectors were compressed by a simple lossless encoder based on JPEG-2000:

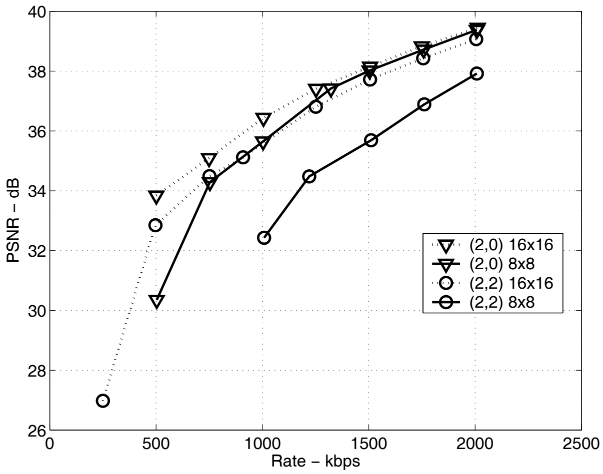

Fig. 1. PSNR [dB] for sequence "Foreman" (CIF) with 3 temporal decomposition levels (GOP size: 8 frames) and lossless compression of motion vectors

this choice results in a high motion bitrate, thus in a high overall bitrate, but still allows a performance comparison between the transforms.

In Fig. 1, we show PSNR versus total bit-rate, including motion, for "Foreman". We present results for two motion block sizes: 16×16 (dashed curves) and 8×8 (solid curves). The (2,0) transform outperforms the (2,2) transform for 16×16 blocks by a significant margin because of the higher motion bit-rate needed in the latter one. This is due to the fact that, since motion is neither invertible nor additive (block-based), two vector fields are needed for the (2,0) transform ($\mathbf{v}_{2k+1\to 2k}$, $\mathbf{v}_{2k+1\to 2k+2}$) while four vector fields are needed for the (2,2) transform ($\mathbf{v}_{2k+1\to 2k}$, $\mathbf{v}_{2k+1\to 2k+2}$, $\mathbf{v}_{2k\to 2k-1}$, $\mathbf{v}_{2k\to 2k+1}$). However, for invertible/additive motion models the motion-rate penalty will disappear [10].

Interestingly, the performance gain of the (2,0) transform over the (2,2) transform is even larger for 8×8 motion blocks. This is not surprising since the motion penalty afflicting the (2,2) scheme is twice larger than in the (2,0) scheme. Moreover, motion accuracy plays an important role in the (N,0) transforms, because they closely approximate their (N, M) lifting counterparts only if $\epsilon_{i+j}(i)[\mathbf{m}] \ll x_{i+j}[\mathbf{m}]$ holds, and this happens only if motion-compensated prediction works efficiently.

5. CONCLUSION AND PERSPECTIVES

We showed that certain trade-offs are unavoidable in lifting-based temporal wavelet transform. We considered modifying the wavelet transform itself and we found out that the (2,0) lifting transform is a very good alternative to the usual 5-3 transform for block-based motion (non-invertible, non-additive). This is an interesting result since today the only real-time method for motion estimation at video rates and ITU.R-601 resolutions is block matching. We plan to study these issues for the (4,0) lifting transform as well.

6. REFERENCES

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

[2] S.-J. Choi and J.W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. on Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[3] B.-J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Proc. Data Compression Conference*, Mar. 1997, pp. 251–260.

[4] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. on Image Process.*, vol. 3, no. 5, pp. 572–588, Sept. 1994.

[5] L. Luo, J. Li, S. Li, Z. Zhuang, and Y. Zhang, "Motion compensated lifting wavelet and its application in video coding," in *IEEE Int. Conf. on Multimedia and Expo*, Aug. 2001.

[6] T. Wiegand and B. Girod, *Multi-frame Motion-Compensated Prediction for Video Transmission*, Kluwer Academic Publisher, 2001.

[7] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, 2001, pp. 1793–1796.

[8] A. Secker and D. Taubman, "Motion compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proc. IEEE Int. Conf. Image Processing*, 2001.

[9] J. Viéron, C. Guillemot, and S. Pateux, "Motion compensated 2D+t wavelet analysis for low rate FGS video compression," in *Proc. Tyrrhenian Int. Workshop on Digital Communications*, Sept. 2002.

[10] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Trans. Image Process.*, 2003 (to appear).

[11] J. Konrad, "Transversal versus lifting approach to motion-compensated temporal discrete wavelet transform of image sequences: equivalence and tradeoffs," in *Proc. SPIE Visual Communications and Image Process.*, Jan. 2004.

[12] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

[13] C. Parisot, M. Antonini, and M. Barlaud, "Motion compensated scan based wavelet transform for video coding," in *Int. Workshop on Digital Communications*, 2002.

[14] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl. 4*, pp. 245–267, 1998.

[15] 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, 1998.

[16] C. Parisot, M. Antonini, and M. Barlaud, "3D scan based wavelet transform and quality control for video coding," *EURASIP Journal on Applied Signal Processing, Special Issue on Multimedia Signal Processing*, Jan. 2003.