EVALUATION OF WAVELET DOMAIN BLOCK MOTION COMPENSATION (WBMC)

Rade Kutil

Salzburg University, Dept. of Scientific Computing, Austria
rkutil@cosy.sbg.ac.at
http://www.cosy.sbg.ac.at/sc/staff/rade.kutil.html

ABSTRACT

Wavelet based image compression has proved its superiority over DCT based methods. However, wavelet based techniques did not have so much success in video coding because motion compensation usually produces residuals that are not suitable for wavelet compression. This paper discusses a motion compensation technique which should fit naturally into wavelet based compression schemes. The motion compensation is solely based on blocks of wavelet coefficients in order to reduce "wavelet-unfriendly" blocking artefacts. This involves wavelet coefficients to be shifted spatially in the wavelet domain.
Keywords: wavelets, motion compensation, visual motion

1. INTRODUCTION

Wavelet based compression has proved its superiority over discrete cosine transform (DCT) based methods in image coding. Therefore, it can be found in many state-of-the-art compression algorithms [10] and image coding standards [2].

However, wavelet based techniques did not have so much success in video coding. The problem is motion compensation which produces prediction residuals that are not quite suitable for wavelet compression. There are approaches for video coding which do not involve traditional motion compensation. [13, 3, 14] are based on 3-D wavelet transform. [8, 15] apply 1-D wavelet transform along motion threads followed by 2-D frame-wise wavelet transform.

However, motion compensation still seems to be the most practicable technique for video coding. [7] proposes an approach to perform motion compensation in the wavelet domain by using undecimated high-pass sub-bands. This approach is a loose approximation of motion compensation, though. [16] proposes a method of hierarchical backward motion compensation that predicts motion and high-pass coefficients from coarse scale approximations. This method has the advantage that no motion vectors have to be encoded.

The author was partly supported by the Austrian Science Fund FWF, project no. P13903.

This paper proposes a new technique of block motion compensation in the wavelet domain (WBMC). Groups of wavelet coefficients that are located within a spacial block are combined to wavelet blocks (WB). A method to shift such a block by arbitrary vectors directly in the wavelet domain is developed. By this it is possible to apply conventional block motion estimation/compensation techniques in the wavelet domain.

The method is compared to conventional block motion compensation techniques in terms of prediction quality, compression performance and computational complexity.

1.1. Conventional Block Motion Compensation

In video coding, frames are predicted from previous frames by dividing the frames into square blocks of equal size (16×16 is common) and assigning a motion vector (MV) to each block. A pixel from the previous (reference) frame with a displacement equal to the MV serves as prediction for a given pixel in the current frame. This is called block motion compensation (BMC) and is used in most of the common video coding standards [1, 5].

Unfortunately, BMC introduces discontinuities at block borders. While this is not a major problem for DCT based compression because DCT is usually also block based and shares block borders with BMC, wavelet based compression techniques suffer seriously under high frequency components caused by block border discontinuities.

Overlapped block motion compensation (OBMC) [9, 12] is a good solution to these problems because it not only increases prediction accuracy but also avoids blocking artifacts. Blocks are usually twice as big in each dimension and overlap quadrant-wise with all 8 neighbouring blocks. Thus, each pixel belongs to 4 blocks. There are 4 predictions for each pixel which are summed up to a weighted mean. For this purpose, blocks are associated with a window function that owns the property that the sum of 4 overlapped windows is equal to 1 everywhere.

The major disadvantages of OBMC are increased computational complexity and the fact that prediction errors and, thus, also the optimal MVs depend on neighbouring blocks and MVs. Therefore, there is no algorithm with polynomial complexity that guarantees optimal MVs. However, there are near-optimal iterative [9] and non-iterative [11] methods with acceptable computational complexity.

Motion estimation (BME, OBME) is the process of finding optimal or near-optimal MVs which minimise the overall prediction error. The prediction error of a block is defined as the mean squared error (MSE) between predicted and actual pixel values over all pixels of the block. For OBME the pixel-wise prediction errors of a block and its overlapping neighbouring blocks have to be weighted according to the window function. As in the process of successively finding/refining MVs some neighbouring MVs are not known yet, the corresponding prediction errors can be ignored (not added) as a sub-optimal solution.

To find optimal MVs, one basically has to calculate the block prediction error for each MV within a certain search range and pick the one with the smallest error (full search). A faster and sub-optimal method is to use a coarse search grid for a first approximation and to refine the grid in the surrounding of this approximation in further steps. The most common representative of this method is the 3-step search which uses search grids of 3×3 MVs and 3 refinement steps to get an overall search range of 15×15.

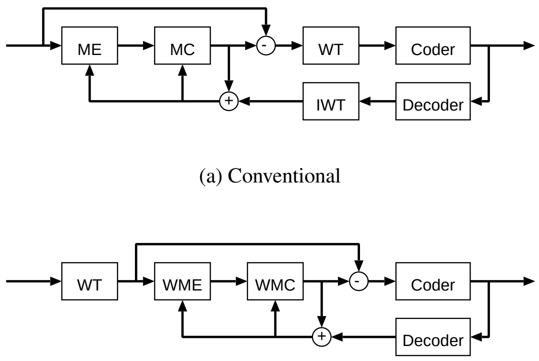
1.2. Motion Compensation in the Wavelet Domain

To fully eliminate blocking artifacts that disturb wavelet based compression, it is an ideal solution to assign a MV to each wavelet coefficient. If it is possible to predict a wavelet coefficient from this MV and the reference frame, the prediction does not include any blocking artifacts that usually result from motion compensation. A brute force method to get predicted wavelet coefficients would be to shift a sufficiently large part of the image (surrounding the corresponding wavelet) and to perform a wavelet transform on this part. This approach is clearly too time consuming and is, therefore, dropped immediately.

This paper proposes a method to predict wavelets by a linear combination of neighbouring coefficients. Thus, a limited number of multiplications is needed per coefficient. In this case, "neighbouring" means neighbouring in space as well as in frequency, i.e. also coefficients in other sub-bands serve as source for coefficient prediction.

Of course, not every single wavelet coefficient gets a distinct MV because this would mean one MV per pixel. Therefore, wavelet coefficients have to be grouped together to form blocks (WBs) just as pixels are in conventional BMC. Each WB gets its own MV. This makes conventional motion estimation techniques applicable to the wavelet domain.

As the borders of a WB are not sharp in the pixel domain and wavelets located at the border of blocks overlap with wavelets from neighbouring blocks, WBs can be interpreted as special overlapped windowed blocks as used



(a) Conventional

(b) Based on MC in the wavelet domain

Fig. 1. Schematic data flow of video encoder

in OBMC. An important difference to overlapped blocks is the fact that the prediction error of WBs does not depend on neighbouring blocks/MVs. The reason for this is that the MSE is wavelet transform invariant for orthogonal wavelets (Parseval's equality) and approximately invariant for biorthogonal wavelets. Therefore, minimising the prediction error of single wavelet coefficients also minimises the MSE of the reconstructed image. Thus, it is sufficient to minimise the MSE of a given WB (calculated in the wavelet domain) independently of neighbouring blocks.

Another reason to perform motion compensation and estimation in the wavelet domain is shown in Figure 1. Usually, the motion prediction of a video coder is based on a decoded reference frame because this is the best that the decoder has and the difference between the predicted and the actual frame should be the same in the encoder and the decoder. Thus, the encoder has to decompress the frame it has just compressed in order to make it available for the subsequent frame prediction. As the wavelet transform is usually the most time consuming part in image decompression, it is advantageous if we can avoid wavelet transforms. If the motion compensation is performed in the wavelet domain, the inverse wavelet transform (IWT) of the reference frame is not necessary. In this way, the encoder complexity can be reduced.

2. SHIFTING WAVELETS

As part of the wavelet decomposition, sub-bands are scaled down by a factor of 2 in each decomposition step. Accordingly, a sub-band at decomposition level l is down-scaled by 2^l. Now, motion compensation in the wavelet domain involves moving wavelet coefficients by arbitrary vectors. It is easy to shift a wavelet coefficient at level l by a vector whose coordinates are multiples of 2^l because such a vec-

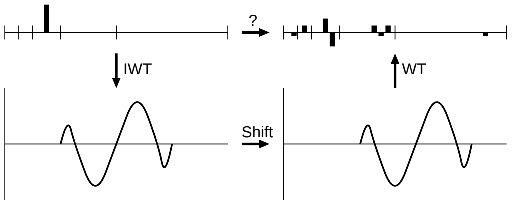


Fig. 2. Shifting 1-D wavelets

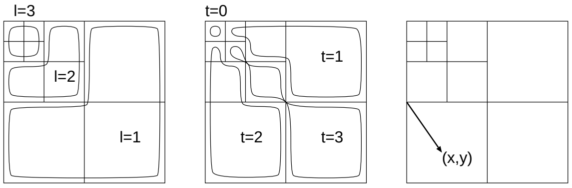


Fig. 3. Locating wavelet coefficients

tor can be down-scaled by dividing it by 2^l. The coefficient is then simply shifted by the down-scaled vector within its sub-band.

Unfortunately, vectors without this property are more complicated to handle as is demonstrated in Figure 2 for the 1-D case. A shift by an arbitrary vector causes many neighbouring coefficients even in other sub-bands to be affected by a given coefficient. Its energy is spread over these coefficients. To incorporate this operation in the wavelet domain, the factors by which neighbouring coefficients are affected have to be known. For this purpose, we introduce a shift operator consisting of a set of such factors for all necessary shift vectors, decomposition levels and sub-band types.

First, we have to find a clear notation for coefficient positions. A coefficient can be located by the decomposition level l, the sub-band type t of the sub-band it is part of, and its position (x, y) within the sub-band (see Figure 3). Thus, (l, t, x, y) is a fully qualified coefficient position. Now, let $\chi_{(l,t,x,y)}$ be the set of wavelet coefficients that is 1 in (l, t, x, y) and 0 everywhere else. $\lambda_{a,b}$ be the shift of an image by the vector (a, b). A shift of an image in the wavelet domain can then be achieved by the concatenation $\mathrm{WT} \circ \lambda_{a,b} \circ \mathrm{IWT}$ of operations, i.e. inverse wavelet transform (IWT) to get the original image followed by a pixelwise shift and a wavelet transform (WT) to change to the wavelet domain again. We can define:

$$\begin{aligned} &M_{a,b}(l, t, x, y, l', t', x', y') \\ &\qquad := (\mathrm{WT}\, \lambda_{a,b}\, \mathrm{IWT}\, \chi_{(l,t,x,y)})(l', t', x', y') \,. \end{aligned}$$

$M_{a,b}$ holds all factors necessary to shift a wavelet coefficient directly in the wavelet domain by a vector (a, b). We can use $M_{a,b}$ to calculate a shift of a set C of wavelet coefficients. This shift is defined as $\mathrm{WT}\, \lambda_{-a,-b}\, \mathrm{IWT}\, C$. The shift of C in the wavelet domain can be calculated as:

$$\begin{aligned} &(\mathrm{WT}\, \lambda_{-a,-b}\, \mathrm{IWT}\, C)(l, t, x, y) \\ &\quad = \sum_{l',t',x',y'} M_{a,b}(l, t, x, y, l', t', x', y') C(l', t', x', y') \,. \end{aligned}$$

However, this is still not satisfying from a computational point of view for two reasons: First, $M_{a,b}$ has a size of N^2 (where N is the image size) which makes it unusable for video coding. Second, a summation over all coefficients (l', t', x', y') is too time consuming. Fortunately, $M_{a,b}$ owns some properties that are capable of reducing computational and memory requirements drastically.

1. $M_{a+p2^{l'},b+q2^{l'}}(l, t, x, y, l', t', x', y')$
   $= M_{a,b}(l, t, x, y, l', t', x' - p, y' - q)$
   This property assures that it is sufficient to keep a and b in the interval $0 \ldots 2^n - 1$ (where n is maximum decomposition depth). If e.g. $a \geq 2^n$ then a can be set to $a \mod 2^n$ plus a multiple of 2^n which can be down-scaled to the sub-band (l', t').

2. $M_{a,b}(l - 1, t, x, y, l', t', x', y')$
   $= M_{2a,2b}(l, t, x, y, l', t', x', y')$
   This property assures that it is sufficient to keep only factors for coefficients at maximum decomposition level $l = n$. For lower levels, the shift operation can be reduced to a scaled shift at level n.

3. $M_{a,b}(l, t, x + p, y + q, l', t', x', y')$
   $= M_{a,b}(l, t, x, y, l', t', x' + p2^{l-l'}, y' + q2^{l-l'})$
   This property assures that it is sufficient to limit x and y to 0 for $l' \leq l$ and to $0 \ldots 2^{l'-l} - 1$ else. All other factors can be accessed through properly scaled shifts of (x', y').

4. $M_{a,b}$ is sparse. Therefore, it is advisable to store elements of $M_{a,b}$ in a set of lists indexed by a, b, t, x, y and l'. l is not needed as index because it can be kept equal to n (see above). l' is introduced as index although it is variable. This is to be able to restrict the range of x and y which depends on l'. Therefore, l' has to be varied within a sufficient range when accessing $M_{a,b}$.

5. Computational and memory requirements can further be reduced by discarding elements below a threshold. This reduces list sizes and accelerates calculation. However, it also introduces errors in the results.

After taking these properties into account, the handling of $M_{a,b}$ and, thus, applying shift operations in the wavelet domain is a feasible process. As we will see, it can be used in wavelet domain based block motion compensation.

(a) Wavelet Block (1,2) in a depth 3 decomposition of a 32×32 image

(b) Reconstruction of a wavelet block

Fig. 4. Example wavelet blocks

3. WAVELET BLOCKS

The idea behind BMC is that a group of pixels is combined to form a block and to assign a single MV to the whole block. This reduces the bit-rate for the encoding of MVs. If we want to apply motion compensation in the wavelet domain, we have to do something similar. We define a wavelet block (WB) as

$$B(x_0, y_0) := \{\, (l, t, x_0 2^{n-l} + x, y_0 2^{n-l} + y) \mid \\ \mid 1 \leq l \leq n,\ 0 \leq x, y < 2^{n-l} \,\} \ .$$

In this way, the block resolution is equal to the resolution of the sub-band at maximum decomposition level, i.e. a WB has a size of 1 coefficient at maximum decomposition level. A WB is visualised in Figure 4.

To predict a WB from a reference frame, the following calculation has to be executed:

$$\begin{aligned} \tilde{C}(B(x_0, y_0)) &:= (\mathrm{WT}\, \lambda_{-a,-b}\, \mathrm{IWT}\, C)(B(x_0, y_0)) \\ &= \left(\sum_{p'} M_{a,b}(p, p') C(p') \right)_{p \in B(x_0, y_0)} . \end{aligned}$$

If C_k are the coefficients of frame k, then the prediction of $C_k(B(x_0, y_0))$ is $\tilde{C}_{k-1}(x_0, y_0)$ and the prediction error is $\mathrm{MSE}(C_k(B(x_0, y_0)) - \tilde{C}_{k-1}(x_0, y_0))$. The task of wavelet block motion estimation (WBME) is to find the MV (a, b) that minimises the prediction error. All conventional BME methods (full search, 3-step search and iterative methods) can be applied in WBME.

In this paper full search and m-step search is used (search range $\pm(2^m - 1)$). In the r-th step the search grid has a granularity of 2^{m-r}. Therefore, it is not necessary to calculate prediction errors for coefficients at a level $l \leq m - r$ because these coefficients contain high-frequency components that cannot be perceived by such a coarse search grid. Note the relationship to the sampling theorem (Nyquist, Shannon). Experiments verify this assumption. It decreases the complexity of the m-step search in WBME.

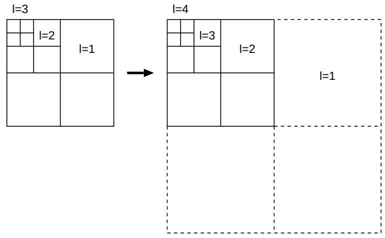


Fig. 5. How to get half-pixel precision in WBMC

4. HALF-PIXEL PRECISION

Half-pixel precision (i.e. motion in $\frac{1}{2}$ pixels) can be achieved easily in WBMC. As each coefficient can be shifted by an arbitrary vector, it can also be shifted by a half pixel. All we need is the proper factors for $M_{a,b}$.

To achieve this formally, we extend virtually the sub-band structure by one high-frequency level (see Figure 5). In this way, the level of each sub-band is increased by 1. Therefore, the range of shift vectors doubles and has to be interpreted in $\frac{1}{2}$ pixel steps.

The coefficients of the virtual sub-bands at level 1 are supposed to be 0 before and after the shift. Therefore, they are not included in any calculation and do not increase computational or memory demands.

5. EXPERIMENTAL RESULTS

The first 100 frames of the "foreman" video sequence are used for experiments throughout this section. Other video sequences have been tested, with very similar results. The frames of this sequence are of the size 176×144 pixels. The graphs in this section show the quality of the prediction of the k-th frame using the $k-1$-th frame as reference frame. Experiments were conducted on a 1 GHz Pentium III.

WBMC is compared to conventional BMC and OBMC. For OBMC the motion estimation method *RAST* [11] is used which is non-iterative. It employs a raster-scan on blocks and includes all neighbouring blocks that have previously been accessed in the scan into the calculation of the overlapped prediction error. A bilinear window function is used.

The search range for MVs is $\{-15 \ldots 15\} \times \{-15 \ldots 15\}$. Thus, the m-step search needs 4 steps. This corresponds to the wavelet decomposition depth 4.

First, let us have a look at the execution times of the used algorithms (see Figure 6). Unfortunately, WBME seems to be slow, especially for a low threshold for the shift factors $M_{a,b}$. Having a look at Figure 7 tells us that a low thresh-

	full search	4-step
BME	0.48	0.06
OBME (RAST)	6.70	0.84
WBME (threshold 0.1)	11.60	0.58
WBME (threshold 0.01)	44.50	2.40
WBME (thr. 0.1, half-pixel)	12.00	1.00
WBME (thr. 0.01, half-pixel)	46.00	4.50

Fig. 6. Timing of wavelet block motion estimation (in seconds)

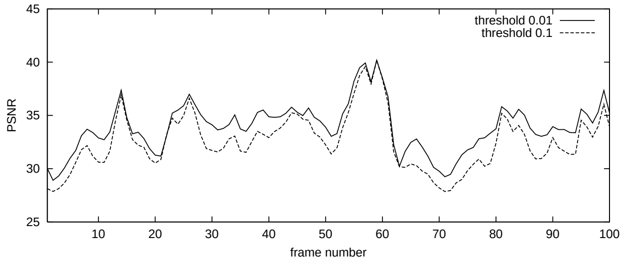


Fig. 7. Influence of the shift factor threshold on the prediction PSNR

old is necessary to achieve good prediction quality. This increases the number of factors used in shift operations, which is the reason for higher execution times.

Figure 8 compares the quality of the three motion compensation methods for the full search motion estimation. Unfortunately, WBMC loses again. However, the advantage of WBMC should be that residuals can be compressed more easily by a wavelet codec. Figure 9 shows similar results for the 4-step search. Note that the quality difference between full search and 4-step search is not significant. Therefore, we will use 4-step search from now on.

Results for half-pixel precision are much better, as one can see in Figure 10. Therefore, we will use half-pixel precision from now on.

To show the performance of a wavelet image codec when used on WBMC residuals, we use the SMAWZ codec [6]. The difference between an actual frame and a predicted frame is encoded at a certain bit-rate. It is then decoded again and

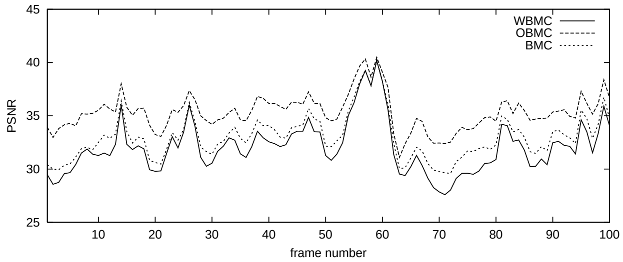


Fig. 8. Prediction quality for full search of MVs

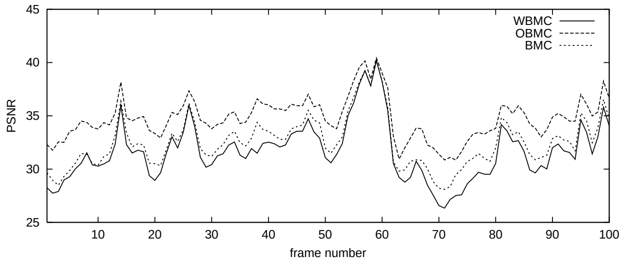


Fig. 9. Prediction quality for 4-step search

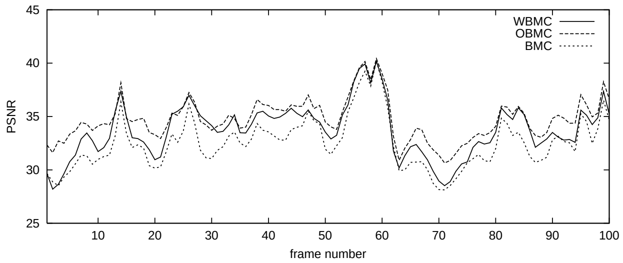


Fig. 10. Prediction quality with half-pixel precision (4-step search)

added to the predicted frame. The result is compared to the actual frame.

Figure 11 and Figure 12 show the results for bit-rates of 0.1 and 0.5 bpp. Very often the PSNR of WBMC is higher than the PSNR of OBMC for frames where it was lower before encoding the residuals. This shows that WBMC residuals can be encoded more efficiently by a wavelet codec than OBMC residuals.

6. FUTURE WORK

There are modified wavelet transforms [4] with better shift invariance. This should make the handling of shift operations in the wavelet domain easier and faster. There exist motion estimation techniques based on wavelets which could be used to find MVs. The hierarchical scan order of m-step search is very similar to the way zero-tree image

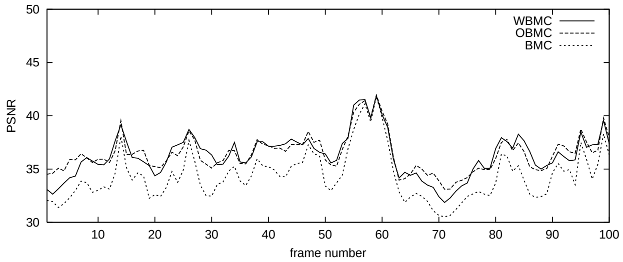


Fig. 11. Coding performance of residuals at 0.1 bpp

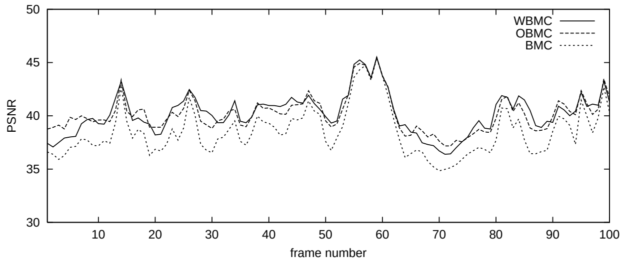


Fig. 12. Coding performance of residuals at 0.5 bpp

codecs work. They could be combined to a video codec that integrates motion compensation and residual encoding into an embedded bit-stream.

7. CONCLUSIONS

A novel motion compensation method (WBMC) was presented. The motion compensation is solely based on blocks of wavelet coefficients. For this purpose, a method to shift wavelet coefficients directly in the wavelet domain was developed.

Unfortunately, WBMC cannot achieve the timing performance and prediction accuracy of overlapped motion compensation. However, an easy method to include half-pixel precision can be applied, which makes WBMC competitive. Moreover, experiments show that the prediction residuals are better suitable for wavelet based compression than residuals of conventional motion compensation are.

8. REFERENCES

[1] L. Chiariglione-Convenor. MPEG-2: Generic coding of moving pictures and associated audio information. ISO/IEC JTC1/SC29/WG11, July 1996.

[2] ISO/IEC JPEG committee. JPEG 2000 image coding system — ISO/IEC 15444-1:2000, December 2000.

[3] B.J. Kim and W.A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In *Proceedings Data Compression Conference (DCC'97)*, pages 251–259. IEEE Computer Society Press, March 1997.

[4] Nick G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253, May 2001.

[5] R. Koenen. Overview of the MPEG-4 standard. ISO/IEC JTC1/SC29/WG11, July 1996.

[6] R. Kutil. A significance map based adaptive wavelet zerotree codec (SMAWZ). In S. Panchanathan, V. Bove, and S.I. Sudharsanan, editors, *Media Processors 2002*, volume 4674 of *SPIE Proceedings*, pages 61–71, January 2002.

[7] F.G. Meyer, A.Z. Averbuch, and R.R. Coifman. Motion compensation of wavelet coefficients for very low bit rate coding. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, October 1997.

[8] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Transactions on Image Processing*, 3(5):559–571, September 1994.

[9] M.T. Orchard and G.J. Sullivan. Overlapped block motion compensation: A estimation-theoretic approach. *IEEE Transactions on Image Processing*, 5:693–699, September 1994.

[10] Amir Said and William 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*, 6(3):243–249, June 1996.

[11] J.K. Su and R.M. Mersereau. Non-iterative rate-constrained motion estimation for OBMC. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, pages II:33–xx, Santa Barbara, CA, USA, October 1997.

[12] J.K. Su and R.M. Mersereau. Motion estimation methods for overlapped block motion compensation. *IEEE Transactions on Image Processing*, 9(9):1509–1521, September 2000.

[13] D. Taubman and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing*, 5(3):572–588, September 1993.

[14] A. Wang, Z. Xiong, P. A. Chou, and S. Mehrotra. Three-dimensional wavelet coding of video with global motion compensation. In *Proceedings Data Compression Conference (DCC'99)*, pages 404–413, Snowbird, UT, March 1999.

[15] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang. Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT). *Applied and Computational Harmonic Analysis*, 10(3), May 2001.

[16] X. Yang and K. Ramchandran. Hierarchical backward motion compensation for wavelet video coding using optimized interpolation filters. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, CA, USA, October 1997.