ON THE IMPORTANCE OF MOTION INVERTIBILITY IN MCTF/DWT VIDEO CODING

Nikola Božinović, Janusz Konrad, Wei Zhao

Deparment of Electrical and Computer Eng.
Boston University
Boston, MA 02215

Carlos Vázquez

INRS-EMT
Université du Québec
Montréal, QC H5A 1K6

ABSTRACT

Motion-compensated temporal filtering implemented using lifting is an effective and efficient temporal decomposition tool that facilitates video compression competitive with the current standards. As recently shown, however, in order that a lifting-based motion-compensated discrete wavelet transform indeed implement the intended filtering along motion trajectories, motion transformation must be invertible and motion composition between frames must be well-defined. A departure from these conditions results in the application of sub-optimal subband decomposition filters which, in turn, degrades coding performance, even if prediction-step energy is minimized during motion estimation. In this paper, we study the impact of motion field invertibility error on the coding performance of an MCTF/DWT video coder. We propose two new motion field inversion methods and compare them to previously reported inversion techniques. We also compare coding results for all inversion algorithms with those of coding based on triangular meshes that are inherently invertible. Our results show that a significant improvement in coding performance is possible with more accurate motion field inversion.

1. INTRODUCTION

Lifting implementations of the discrete wavelet transform (DWT) have been used extensively by the image and video processing community; they allow fast and memory-efficient implementation of the transversal (standard) wavelet filtering [1]. Recently, lifting has been incorporated into motion-compensated temporal filtering (MCTF) in 3D-DWT video coders [2, 3]. It is well-known that perfect reconstruction is an inherent property of the lifting structure, even if the input samples undergo non-linear operations, such as motion compensation [4, 3]. However, in order for a lifting structure to exactly implement the original transversal wavelet filtering, motion transformation must be invertible (Haar DWT) or motion composition must be well-defined (higher-order DWTs) [5].

In general, implementation of one stage of MCTF requires one forward and one backward motion field referenced at each frame (the exception being the Haar DWT, where only half of the motion fields are needed). However, independently-estimated motion fields between two subsequent frames, one mapping frame $2k+1$ to frame $2k$ and the other mapping frame $2k$ to $2k+1$, are not, in general, inverses of each other, which can lead to decrease in the coding performance. From the early days of 3D-DWT coding, methods were sought to compute a forward motion field from the transmitted backward field (or vice versa), in an attempt to reduce the amount of motion information to be transmitted. For the most popular motion model based on translating blocks, it was soon discovered that its use in MCTF introduces the appearance of the so-called "disconnected" pixels [6] that occur in areas not conforming to the rigid translational motion model (e.g., expansion, contraction, rotation), and in occluded/newly-exposed areas. This, in turn, leads to ambiguity in selecting the backward motion field required for the update lifting step. As a solution to this problem, ad-hoc methods [7] as well as more recent distortion-model-based techniques [8, 9] were proposed, suggesting various ways of finding the optimal "update" step (i.e., deriving inverse motion from a given motion field). While these methods search for the optimal update (for a given prediction) in terms of the reconstruction error, none of them investigates physical properties and relations between motion fields involved in both prediction and update steps.

In parallel to these efforts, deformable-mesh motion models have been proposed for MCTF/DWT video coding [3]. Unlike the traditional block-matching, which assumes rigid translation of each block, mesh-based models permit the use of affine mapping (triangular mesh) and bilinear transformation (quadrilateral topology). Since deformable-mesh motion models are invertible and since motion composition is well-defined (both under conditions of preserving mesh connectivity), MCTF based on these models results in exact temporal subband decomposition; the invertibility of mesh-based motion doesn't allow for existence of the aforementioned "disconnected" pixels. Also, the composition of motion fields estimated at different levels of temporal decomposition permits a compact representation of motion fields, regardless of the temporal support of a particular DWT used. On the downside, mesh models suffer from strong regularization required to preserve mesh connectivity resulting in excessively smooth motion fields and, thus, reduced performance in occlusion areas. Also, mesh-based motion estimation, typically implemented through iterative hierarchical hexagonal refinement [10], is computationally very complex (typically more than tenfold compared to block matching). Recently, more efficient methods for the use of triangular meshes without incurring the high computational cost of hexagonal refinement were proposed [11].

In this paper, we investigate block-based and mesh-based motion estimation, and for block motion models we analyze various motion inversion techniques in the context of MCTF/DWT. We propose two new motion field inversion methods, one based on nearest-neighbor interpolation and one that uses spline-based approximation. Both methods are applicable to arbitrarily-derived vector fields (not necessarily block-based) of arbitrary precision (not necessarily full-pixel). We show encouraging experimental results for block-based motion model.

This work was supported by the National Science Foundation under grant CCR-0209055. [nikolab,jkonrad]@bu.edu, vazquez@inrs-emt.uquebec.ca

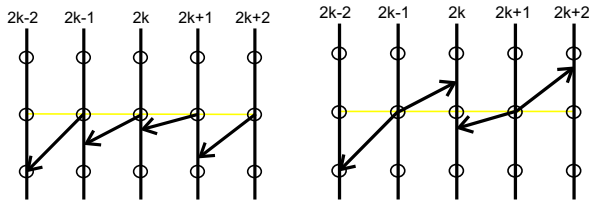


Fig. 1. Motion estimation for MCTF: a) unidirectional; b) bidirectional.

2. UNIDIRECTIONAL VS. BIDIRECTIONAL MOTION ESTIMATION

In contrast to standard backward-predictive motion estimation used for P frames in hybrid coding (where current frame is always ahead and predicted from "previous" reference frame), the use of longer filters in MCTF (e.g., 5/3 instead of Haar) leads to two alternatives in the way motion estimation is performed: *unidirectional* or *bidirectional* [12]. In the unidirectional case (Fig. 1(a)), just like in predictive coding, motion vectors are always pointing backwards (i.e., the tail of the motion vector is always at pixel position in the current frame, while vector arrowheads are possibly off full-pixel grid in the "earlier" reference frame). This permits the direct use of readily available standard motion estimation algorithms. In the bidirectional case, the estimated motion fields always originate at the frames aligned with the high temporal subband (typically, odd frames), and alternate in pointing backward and forward, as shown in Fig. 1(b). At the first glance, neither method seems advantageous over the other since both minimize the overall prediction error, although defined somewhat differently (backward prediction versus a combination of backward and forward prediction). However, a more careful analysis of MCTF employing filters with longer temporal support reveals a significant difference between them. For example, consider 5/3 filters implemented using lifting; while the bidirectional motion estimation minimizes the energy in the high subband (sum of the forward and backward prediction errors), the unidirectional approach does not. This is so because although one half of motion vectors used in the prediction step is directly computed (by minimizing the backward prediction error), the other half used for prediction must be derived (indirectly calculated) by some form of inversion that, in general, does not minimize the forward prediction error.

3. MOTION FIELD INVERSION

It has been shown recently [5] that invertibility of motion plays a significant role in both lifting and transversal implementations of MCTF. In a transversal implementation, motion needs to be invertible for Haar filters (obey composition property for higher-order filters) in order to assure the perfect reconstruction property. Although in lifting, perfect reconstruction is guaranteed regardless of motion compensation performed, the motion transformation must be invertible (obey composition) in order to implement the intended transversal filters (i.e., Haar, 5/3). If these conditions are not satisfied, lifting implements a suboptimal temporal wavelet decomposition.

In order to implement the prediction and update steps of motion-compensated temporal DWT, both backward and forward motion fields between frame-pairs are needed. The simplest approach is to estimate both motion fields independently (i.e., from frame $2k+1$ to frame $2k$, and from frame $2k$ to frame $2k+1$). Although optimal in terms of the total prediction error, this method requires that both vector fields be transmitted. The other, not very obvious, disadvantage of this approach is that the two fields are not necessarily close to being mutual inverses, which might result in a reduced coding performance when such independently-estimated motion fields are used for the MCTF.

As an alternative to the independent motion field estimation, we can compute only one of the fields (i.e., backward) and estimate the other (forward) field by some sort of inversion. We have earlier reported on two simple methods of motion inversion [13]: a "collinear-extension" motion inverse and a "neighbor-frame-copy" motion inverse. The former technique assumes collinearity between the forward and backward motion vectors originating at the same frame [14], which corresponds to the assumption of constant-velocity motion over three frames. The latter method uses the motion field of a neighboring frame with the sign changed and has two subclasses: one where motion is estimated in the unidirectional fashion and the other with bidirectionally estimated motion, as described above. All three inversion methods are illustrated in Figs. 2(a)-(c). Solid lines represent motion vectors that are directly estimated from input frames using prediction error criterion, while dashed lines show vectors that are obtained through inversion. Similarly, the open arrowheads represent motion vectors used in the prediction step and closed arrowheads denote vectors that are used for the update step. It should be clear from these plots, that out of these three methods only the one that uses bidirectional motion estimation performs optimally in terms of minimizing the high-subband energy.

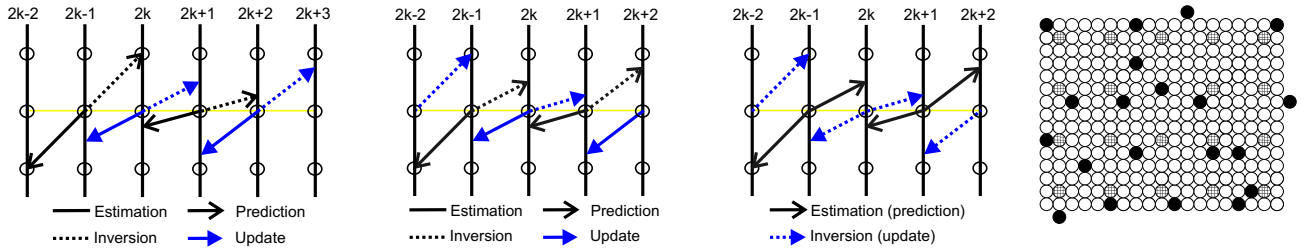
The above techniques compute a very coarse inverse motion field. The proper inversion should project, through motion compensation, all grid points from the reference image to the plane of the target image, and then change the sign of each motion vector. However, the projected grid is irregular and some form of irregular-to-regular data interpolation is needed [15]. This is shown in Fig. 2(d), which is constructed for quarter-pixel motion precision and illustrates the fact that based on the knowledge of motion components at irregular grid points (black) we need to recover motion at the regular grid points (hashed).

3.1. Nearest-neighbor motion inversion

For all motion vectors defined on an irregular grid in the target frame, we compute the vectors (processing one motion coordinate at a time) at regular grid locations using the nearest-neighbor interpolation (Fig. 3(a)). First, each irregular location is mapped to the nearest pixel and the associated motion vector is copied there (pixel becomes "occupied"). Then, all "unoccupied" pixels are scanned successively and assigned the motion vector from the nearest "occupied" pixel. The procedure is repeated until all pixels become "occupied". Details of this procedure can be found in a recent technical report [16].

3.2. Spline-based motion inversion

As the nearest-neighbor interpolation is known to have poor performance, we have also applied an advanced irregular-to-regular interpolation method based on spline approximation [15] illustrated in Fig. 3(b). Although based on cubic splines, this method



(a) (b) (c) (d)

Fig. 2. (a) "Colinear-extension" motion inversion; (b) "neighbor-frame-copy" motion inversion for unidirectional motion estimation; (c) "neighbor-frame-copy" motion inversion for bidirectional motion estimation; and (d) irregular-to-regular motion-field interpolation: interpolation of motion vectors at regular positions (hashed) based on the knowledge of vectors at irregular positions (black).

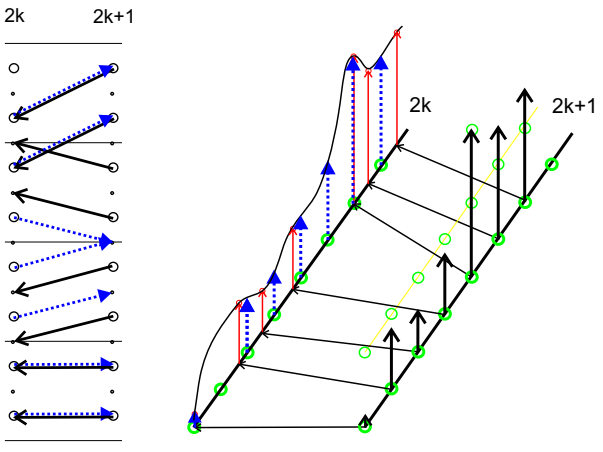
is not an interpolation method since it uses a prior term related to the curvature of the computed surface for each motion component. Due to this prior, the resulting motion-component surface needs not pass through the original data points induced by motion compensated projection. Similarly to the nearest-neighbor, we apply this method twice: once for the x and once for the y component of motion vectors.

3.3. Inversion error

In order to measure the motion field inversion quality objectively, we developed the following invertibility error:

$$\epsilon_d = \sum_{\mathbf{x}} |\mathbf{d}^b(\mathbf{x}) + \tilde{\mathbf{d}}^f(\mathbf{x} + \mathbf{d}^b(\mathbf{x}))|$$

where $\mathbf{d}^b = [d_x^f, d_y^f]^T$ and $\mathbf{d}^f = [d_x^b, d_y^b]^T$ are backward and forward motion vectors, respectively, while $\tilde{\mathbf{d}}$ denotes interpolation (in our case bilinear) of x and y components of $\mathbf{d}$ at non-grid positions. Clearly, this error measures the sum of departures of points in frame $2k+1$ when each of them is projected onto frame $2k$ using the backward motion field and then back projected onto frame $2k+1$ using the (interpolated) forward motion field. A pair of motion fields being perfect inverses of each other would result in zero error ϵ_d.



(a) (b)

Fig. 3. Interpolation used in motion field inversion: (a) nearest neighbor, (b) spline. Solid lines are used for estimated and dashed lines for inverted motion vectors.

4. EXPERIMENTAL RESULTS

Results provided in this section are obtained using CIF resolution *Foreman* and *Coastguard* sequences at 30 fps. The block-based motion estimation is implemented using exhaustive-search block matching at full spatial resolution with search range of $\pm$ 8 pixels per frame with $1/8$-pixel accuracy and using bicubic interpolation of the original frames. We used block size of 16×16 pixels, and the mean-squared error distortion metric.

We compared the various inversion methods to the modified mesh structure developed by us earlier [17]. In the mesh-based approach, node-point motion vectors were estimated using hierarchical hexagonal refinement algorithm initialized with zero-motion field. The search range and motion precision were kept the same for all configurations.

Table 1 shows the PSNR performance for both sequences at the average bit-rate of 500 kbps, with motion bit-rate not included in the overall bit budget. This allows us to analyze the quality of subband decomposition of different methods without the bias introduced by different motion overheads. In this experiment, we have used an implementation of the JPEG2000 image compression standard to intra-code the subbands obtained after a single decomposition level of the motion-compensated 5-3 lifting transform. We can see that virtually any method of motion inversion outperforms two independently estimated motion fields. This suggests that, unlike in hybrid coding schemes, minimization of the prediction error isn't the only significant criterion that needs to be used for the estimation of suitable motion operator. To the contrary, it suggests that the overall coding performance also improves when forward and backward motion fields are "well matched", i.e., closer to being inverses of each another. Along with the coding gains, we also show the invertibility error ϵ_d, introduced earlier. It is clear that there exists a strong correlation between this measure and the coding performance. Note that the forward motion field is identical for all block-based motion models.

In Table 2, we show the PSNR performance for both sequences at the average rate of 1000 kbps (motion rate included). We use the same coder as in the previous experiment but with three temporal decomposition levels of the motion-compensated 5-3 lifting transform. The motion was losslessly encoded and the average overhead for motion information in our experiments ranged from 22.1% to 29.3%, depending on the motion model used. The first row ("5-3 Jnt") shows PSNR performance for two independently-estimated but jointly-coded motion fields [13]. The next five rows show the PSNR obtained through different and progressively more sophisticated techniques of motion inversion. The last row gives

Table 1. PSNR performance [dB] at 500 kbps (motion rate not included)

Configuration	*Coastguard*		*Foreman*	
	PSNR	ϵ_d/pixel	PSNR	ϵ_d/pixel
5-3-Ind	31.07dB	0.27	34.33dB	0.61
5-3-Prv-Uni	(+0.03)	0.24	(+0.03)	0.53
5-3-Col	(+0.04)	0.22	(+0.05)	0.50
5-3-Prv-Bi	(+0.04)	0.20	(+0.06)	0.45
5-3-NN	(+0.07)	0.08	(+0.09)	0.19
5-3-Spline	(+0.13)	0.04	(+0.12)	0.11
5-3-ModMsh	(+0.17)	0	(+0.14)	0

PSNR for the modified triangular-mesh motion field [17].

We notice the increase in PSNR for more accurate inversions of block-based motion fields. Still, the mesh outperforms the best results obtained through inversion by an average of 0.3dB. However, the coding gain of coder configurations utilizing mesh comes at the price of significantly higher computational cost of iterative hexagonal refinement motion estimation.

Table 2. PSNR performance [dB] at 1000 kbps

Configuration	*Coastguard*	*Foreman*
5-3-Jnt	32.42dB	36.71dB
5-3-Prv-Uni	(+1.19)	(+0.73)
5-3-Col	(+1.23)	(+0.72)
5-3-Prv-Bi	(+1.28)	(+0.78)
5-3-NN	(+1.37)	(+0.84)
5-3-Spline	(+1.52)	(+1.03)
5-3-ModMsh	(+1.81)	(+1.36)

5. CONCLUSIONS

We have compared different strategies for motion inversion in the context of wavelet video coding. We showed that motion inversion improves coding performance when compared to independent estimation, even before taking into account larger motion bit-rate in the case of two independently estimated motion fields. This suggests that, unlike in the predictive coding of hybrid schemes, prediction error measure should not be used as the exclusive criterion for estimating motion fields needed for MCTF.

We introduced a quantitative measure of "invertibility" that quantifies the departure from invertibility of a motion field pair. We established a firm correlation between this invertibility error and coding performance. As expected, two advanced inversion methods proposed here outperform previously used trivial inversion methods.

6. REFERENCES

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

[2] B. Pesquet-Popescu and V. Bottreau, "Three-dimensional lifting schemes for motion compensated video compression," in *in Proc. IEEE ICASSP*, 2001, pp. 1793–1796.

[3] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1530–1542, Dec. 2003.

[4] A. Bruekens and A. van den Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE J. Sel. Areas Commun.*, vol. 10, 1992.

[5] 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, vol. 5308.

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

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

[8] B. Girod and S. Han, "Optimum update for motion-compensated lifting," *IEEE Sig. Proc. Lett.*, 2004 (in print).

[9] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar, "Improved update operators for lifting-based motion-compensated temporal filtering," *IEEE Sig. Proc.Lett.*, 2004.

[10] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 339–356, June 1994.

[11] Y. Wang, S. Cui, and J. E. Fowler, "3D video coding using redundant-wavelet multihypothesis and motion-compensated temporal filtering," in *Proc. IEEE Int. Conf. Image Processing*, 2003.

[12] A. Golwelkar, *Motion compensated temporal filtering and motion vector coding using longer filters*, Ph.D. thesis, RPI, ECSE Dept., Sept. 2004.

[13] N. Božinović, J. Konrad, T. André, M. Antonini, and M. Barlaud, "Motion-compensated lifted wavelet video coding: toward optimal motion/transform configuration," in *Signal Process. XII: Theories and Applications (Proc. Twelfth European Signal Process. Conf.)*, Sept. 2004.

[14] V. Valentin, M. Cagnazzo, M. Antonini, and M. Barlaud, "Scalable context-based motion vector coding for video compression," in *IEEE EURASIP Picture Coding Symposium (PCS)*, Apr. 2003.

[15] C. Vázquez, E. Dubois, and J. Konrad, "Reconstruction of irregularly-sampled images in spline spaces," *IEEE Trans. Image Process.*, Jun. 2004 (in print).

[16] W. Zhao, "Motion compensation in temporal discrete wavelet transforms," Tech. Rep. 2004-04, Boston University, Dept. of Electr. and Comp. Eng., Aug. 2004.

[17] N. Božinović and J. Konrad, "Mesh-based motion models for wavelet video coding," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, May 2004, vol. III, pp. 141–144.