

MOTION-COMPENSATED LIFTED WAVELET VIDEO CODING: TOWARD OPTIMAL MOTION/TRANSFORM CONFIGURATION

*Nikola Božinović**, *Janusz Konrad**, *Thomas André#*, *Marc Antonini#*, *Michel Barlaud#*

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

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

ABSTRACT

Various coding schemes based on lifting implementation of the discrete wavelet transform applied along motion trajectories have recently gained a lot of interest in video processing community as strong candidates to succeed current state-of-the-art hybrid coders. Still, there are a number of very important issues, including the choice of particular wavelet transform and motion model, that have significant impact on the overall coding performance and will determine usefulness of this class of coders. In this paper, we classify and discuss different motion/transform configurations that are being used in motion-compensated lifting-based wavelet transforms. Our results show that coder performance changes significantly for different combinations of motion models and transforms used.

1. INTRODUCTION

Lifting implementations of the discrete wavelet transform (DWT) have drawn a lot of attention in the image and video processing community; they allow fast and memory-efficient implementation of the transversal (standard) wavelet filtering [1]. Recently, lifting has been extended to the temporal dimension and, in order to increase subband decomposition efficiency, has been combined with motion compensation [2, 3, 4]. 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 [5, 6, 7]. However, in order for a lifting structure to exactly implement the original transversal wavelet filtering, motion transformation must be invertible for the Haar wavelet while motion composition must be well-defined for other wavelets [8].

In addition to very popular and widely used block-based motion models, deformable-mesh motion models have been proposed for DWT video coding [7]. Applied within motion-compensated lifting framework these models allow efficient temporal subband decomposition along motion trajectories. Moreover, since deformable-mesh motion models are invertible and since motion composition is well-defined for them, motion-compensated lifting based on these models implements exactly the transversal wavelet filtering along motion

trajectories and thus results in exact temporal subband decomposition. Invertibility of mesh-based motion model overcomes many of the problems observed in block-based motion since the existence of unique trajectories (i.e., one-to-one correspondence between all positions in analyzed frames) doesn't allow for appearance of "disconnected" pixels. However, the inherent smoothing of motion fields in mesh-based motion estimation algorithms can have a negative effect on motion compensation effectiveness.

In this paper, we compare block-based and mesh-based motion models in the context of motion-compensated lifted DWT video coding. We perform the comparison on the so-called 1-3 lifting [9, 7, 10] using a single block motion field per frame and 5-3 lifting using two block motion fields per frame or one mesh-based field. We include in this comparison 5-3 lifting with a recently proposed refinement of the standard triangular mesh-based model [11]. We also discuss tradeoffs in rate allocation between texture and motion.

2. MOTION MODELS

The popular block-based motion models, such as one used in MPEG and H.26X standards, would be an obvious choice for use in motion-compensated lifted DWT video coding. Typically, each block undergoes translation, although more complex motion can be used as well (affine, quadratic models). An error metric measuring the quality of matching between blocks in two frames is defined and a minimization algorithm is devised to find the optimal set of parameters. There exists a large body of results concerning the choice of the metric and parameter estimation algorithms. Moreover, over the years a variety of hardware implementations have been developed thus making the block model very suitable for real-time video coding. However, numerous attempts to incorporate this motion model into 3D-DWT coding structure suffered from the appearance of the so-called "disconnected" pixels [12] occurring in areas not conforming to the rigid translational motion model (e.g., expansion, contraction, rotation) and in occluded/exposed areas. These difficulties stem from the fact that translating-block models do not preserve topology of the mesh induced by block vertices; although blocks form a partition of the reference frame (square blocks shown in Fig. 1(a)), this is not the case for the target frame (blocks may overlap).

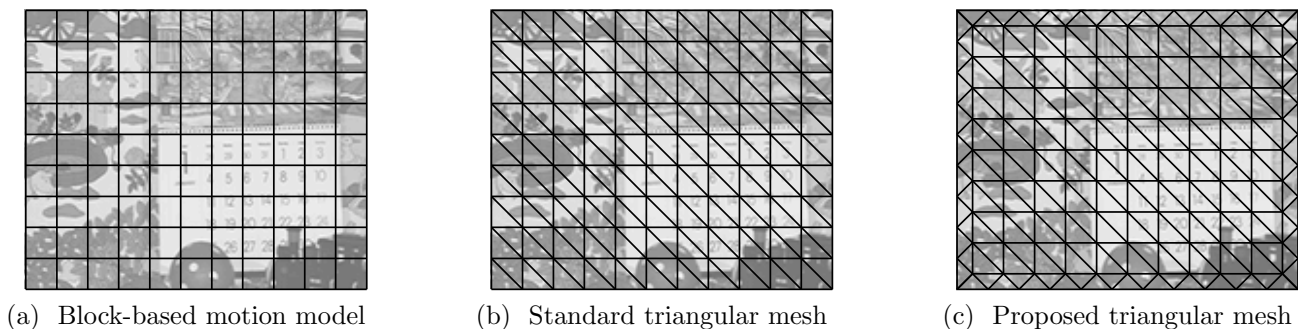


Figure 1: Examples of different node-point topologies.

Recently, mesh-based motion models have been shown to be a good alternative to block-based models. In a regular-mesh case, regular topology is used to partition the reference frame. This mesh is subsequently deformed, by node-point displacements, into another mesh in the target frame. Although topology may not be preserved in a general case, it is possible to constrain node-point movements so that the topology is preserved. Since displacements for points between mesh nodes are interpolated, mesh-based models can account for non-rigid motion. This is unlike block matching where each block is allowed a single displacement thus modeling accurately only rigid-body translational motion.

Although node topologies can be complex, a very successful approach has been to use triangular patches, where, through a suitable model, displacements of three neighboring nodes define displacements anywhere within a triangle. A triangular mesh can be built from the common square-block partitioning thus preserving block positions in the reference frame; mesh nodes can be set at the corners of all square blocks and each block divided in half along its diagonal (Fig. 1(b)). Motion compensation within each mesh element (patch) is typically accomplished by affine spatial transformation whose parameters are computed from node-point motion vectors. This corresponds to a planar interpolation of the horizontal and vertical components of node-point displacements over the whole patch. As the result, affine model assures motion continuity between neighboring patches.

It is well-known that motion estimation between two images fails whenever a given intensity structure exists in one image but not in the other (occluded and newly exposed areas). Unfortunately, since this is common, almost always parts of images have undefined (underlying) motion. In practice, since motion of all image points is needed for compression, an error norm is defined and minimized with respect to some parameters. The computed motion has nothing to do with the true (underlying) motion in this area, but can be used effectively for compression. In mesh-based motion estimation, this is addressed by adapting the mesh topology to image content, however it results in increased computational complexity and significant motion-rate overhead.

A particular case of occluded and newly-exposed areas are image boundaries. Whenever a camera moves and/or objects leave or enter the field of view, features disappear or appear. This results in significant discrepan-

cies between original and predicted frames along the frame boundaries, and leads to a reduced compression efficiency. In order to address this, an improved mesh-based motion compensation was recently proposed [11]. A new triangular mesh topology is created by shifting mesh node-points by half of the inter-node distance toward inside of the frame while constructing a double-density mesh at the frame boundary (Fig. 1(c)). In this way, errors in node-point motion estimates affect fewer pixels than in the standard triangular mesh case.

3. TEMPORAL DWT MODELS

In this paper, we consider temporal 5-3 lifted DWT, and a recently proposed alternative, the so-called 1-3 lifted DWT [9, 7, 10]. The latter transform consists of the same highpass filter as in the regular 5-3 transform but 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 1-3 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 1-3 transform has significant overlap of frequency subbands and thus low-subband wavelet coefficients carry significant amount of information from the high-subband (this can be thought of as additional aliasing due to lack of lowpass filtering). If motion is inaccurately computed, however, error due to imprecise motion compensation in the low-subband of the 5-3 transform may manifest itself as “ghosting”, and may be, in fact, comparable to the additional aliasing error in the 1-3 transform. It remains to be seen at what motion accuracy the two errors are of the same magnitude.

4. MOTION/TRANSFORM CONFIGURATIONS

We are describing below different motion and transform configurations. In the case of 5/3 transform, the various motion models used aim at varying the ration of rate allocated to texture and motion. Table 1 lists the eight configurations for wavelet video coding discussed below. All eight configurations utilize lifting implementation of the wavelet transform.

Table 1: Overview of different coding configurations (MF = motion field, BM = block matching, fr = frame)

Code	Transform	MFs/fr	Motion model	Motion estimation	2×MF coding	Inverse MF
1-3	1-3	1	Block	BM	N/A	N/A
5-3-Ind	5-3	2	Block	BM	Independent	N/A
5-3-Jnt	5-3	2	Block	BM	Joint	N/A
5-3-Prv	5-3	1	Block	BM	N/A	Previous-frame
5-3-Col1	5-3	1	Block	BM	N/A	Collinear
5-3-Col2	5-3	1	Block	Constrained BM	N/A	Collinear
5-3-Msh	5-3	1	Triangular mesh	Hexagonal refin.	N/A	N/A
5-3-ModMsh	5-3	1	Modif. tri. mesh	Hexagonal refin.	N/A	N/A

1-3 lifted DWT

In this case, for each odd frame one forward and one backward block motion field is computed and losslessly encoded (Section 5). No motion fields are needed for even frames.

5-3 lifted DWT with two motion fields coded independently

Forward and backward block motion fields are computed independently in both directions (no collinearity) for each even and odd frame. The motion fields are encoded independently using a lossless encoder.

5-3 lifted DWT with two motion fields coded jointly

Similar to the case above, except that motion fields are encoded jointly. In particular, the forward field is encoded as above and then used as prediction (with sign changed) for the backward field. The resulting prediction error is encoded losslessly.

5-3 lifted DWT with previous-frame inverse MF

In order to reduce the motion rate, only one motion field is transmitted for each frame. However, since in encoding two motion fields are needed, the inverse motion field is approximated by motion field associated with the preceding frame, i.e., for a given block, backward motion vector (inverse) is made equal to the (forward) motion vector (with changed sign) from same-position block in the previous frame [7].

5-3 lifted DWT with collinear inverse MF

Here, again in order to reduce the motion rate, only one motion field is transmitted for each frame. The inverse motion field is recovered by assuming collinearity; for each block, its backward motion vector is assumed to equal its forward motion vector with opposite sign. In a sense, a pseudo-inverse motion field is computed during motion compensation. However, we assume that during motion estimation the forward field is computed without collinearity assumption. This is the case when motion fields used for 1-3 lifted DWT are applied to the 5-3 case under collinearity assumption. The difference between this case and the previous one above, is that

here the current forward motion field is used to compute the backward field, while above it is the previous motion field that is used in this purpose. In principle, the current-frame approach should perform better in case of faster motion.

5-3 lifted DWT with optimized collinear inverse MF

Similar to the above case, except that collinearity is taken into account not only during motion compensation, but also during motion estimation. This is implemented by means of constrained block matching. The resulting motion fields are optimal in the sense that they achieve the best compromise between accurately compensating in the forward and backward directions using one single motion field.

5-3 lifted DWT with triangular mesh

This is the case studied extensively elsewhere [7]. Regular triangular mesh is used while mesh-node displacements are estimated using hexagonal refinement [13].

5-3 lifted DWT with modified triangular mesh

Similar to the above case, except that a modified triangular mesh and a new refinement algorithm are used [11].

5. EXPERIMENTAL RESULTS

Results provided in this section are obtained using SIF-resolution *Mobile & calendar* and *Football* sequences at 30 fps.

The block-based motion estimation is implemented using exhaustive-search block matching at full spatial resolution with search range of ± 8 pixels per frame for the *Mobile & calendar* sequence and ± 16 pixels per frame for the *Football* sequence with 1/8 pixel accuracy and using bicubic interpolation [14] of the original frames. We used block size of 16×16 pixels, and the squared-differences distortion metric.

The standard and modified meshes are created as illustrated in Fig. 1. In both cases, node-point motion vectors were estimated using hierarchical hexagonal refinement algorithm initialized with zero-motion field

[11]. The search range and motion precision were kept the same for all configurations. For motion estimation, in the case of modified mesh topology, we used the strategy that starts motion estimation in the center of the frame, and progressively includes more nodes closer to image boundaries [11].

Table 2 shows the PSNR performance for both sequences at the average rate of 800 kbps. We have used an implementation of the JPEG2000 image compression standard to code each subband obtained with two decomposition levels of the motion-compensated 5-3 lifting transform. The motion was encoded losslessly using JPEG-LS directly on arrays of horizontal and vertical motion components. Average overhead for motion information in our experiments ranged from 22% to 39%, depending on motion model used. Note the slight PSNR gain of the modified mesh over the regular mesh and a more significant one over block motion models. However, coding gain of coder configurations utilizing mesh comes at the price of significantly higher computational cost of iterative hexagonal refinement motion estimation. We can also notice that “collinear inverse” outperforms “previous inverse” with gain being more significant in the sequence with faster motion. Finally, configuration using 1-3 lifted DWT showed very solid coding performance (within 0.3 dB of the best PSNR for “Mobile & Calendar”), at the lowest computational cost.

Table 2: PSNR performance [dB] at average of 800 kbps

Configuration	<i>Mobile & calendar</i>	<i>Football</i>
1-3	26.29	26.57
5-3-Ind	25.99	26.64
5-3-Jnt	26.11	26.72
5-3-Prv	26.32	26.69
5-3-Col1	26.33	26.75
5-3-Col2	26.35	26.83
5-3-Msh	26.42	27.31
5-3-ModMsh	26.56	27.44

6. CONCLUSIONS

We have studied different motion models and configurations in the context of lifting implemented wavelet video coding. In our experiments, mesh-based models slightly outperformed other configurations with the penalty of significantly higher computational cost. In continuation of this work we plan to perform extensive tests for variety of sequences and target bit-rates in order to find optimal motion/transform configuration.

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 *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, 2001, pp. 1793–1796.
- [3] 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, pp. 1029–1032.
- [4] C. Parisot, M. Antonini, and M. Barlaud, “Motion-compensated scan based wavelet transform for video coding,” in *Tyrrhenian International Workshop on Digital Communications*, Sept. 2002.
- [5] A. A. Bruekens and A. W. van den Enden, “New networks for perfect inversion and perfect reconstruction,” *IEEE J. Sel. Areas Commun.*, vol. 10, 1992.
- [6] H. J. Heijmans and J. Goutsias, “Nonlinear multiresolution signal decomposition schemes: Part ii: morphological wavelets,” *IEEE Trans. Image Process.*, vol. 9, pp. 1897–1913, Nov. 2000.
- [7] A. Secker and D. Taubman, “Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression,” *IEEE Trans. Image Process.*, 2004 (to appear).
- [8] 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.
- [9] 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.
- [10] T. André, M. Cagnazzo, M. Antonini, M. Barlaud, N. Božinović, and J. Konrad, “(N,0) motion-compensated lifting-based wavelet transform,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, May 2004 (to appear).
- [11] N. Božinović and J. Konrad, “Mesh-based motion models for wavelet video coding,” in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, May 2004 (to appear).
- [12] J.R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 559–571, Sept. 1994.
- [13] Y. Nakaya and H. Harashima, “Motion compensation based on spatial transformations,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, June 1994.
- [14] R.G. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.