MESH-BASED MOTION MODELS FOR WAVELET VIDEO CODING

Nikola Božinović and Janusz Konrad

Department of Electrical and Computer Engineering, Boston University
8 St. Mary St., Boston, MA 02215

ABSTRACT

Discrete wavelet transforms implemented using lifting along motion trajectories are effective and efficient temporal decomposition tools that facilitate video compression competitive with the current standards. As recently shown, however, in order that a lifting-based motion-compensated wavelet transform be equivalent to its transversal (standard) implementation, motion transformation must be invertible and motion composition between frames must be well-defined. In this paper, we discuss various mesh-based motion models that satisfy requirements of invertibility and composition, and thus are suitable for use in motion-compensated lifting-based wavelet transforms. We propose a new mesh configuration that preserves regularity of the mesh structure but provides better motion compensation compared to previously-reported mesh topologies, particularly in the proximity of image boundaries. Our results show that an improvement in motion compensation and overall compression performance is possible with only a fractional increase in motion overhead bit-rate.

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]. 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 [3, 4, 2]. However, in order for a lifting structure to exactly implement the original transversal wavelet filtering, motion transformation must be invertible and motion composition must be well-defined [5].

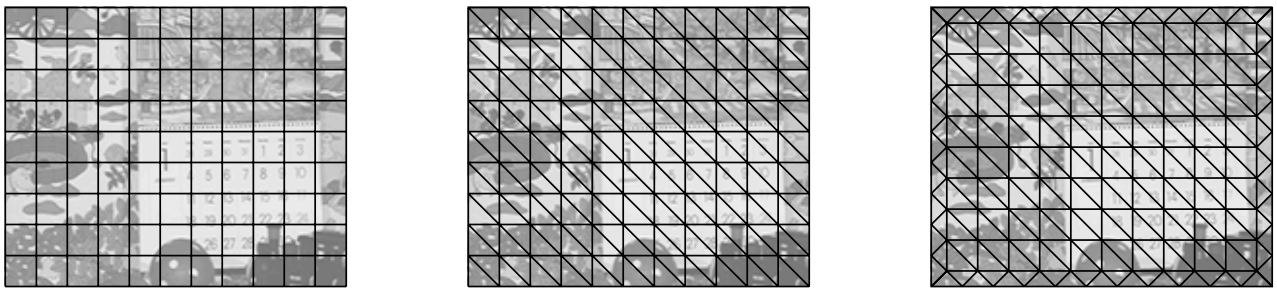
An obvious motion model to use in motion-compensated video coding would be a block-based model (used in MPEG and H.26X); each block undergoes rigid motion, typically translation. However, attempts to incorporate this motion model into 3D-DWT coding structure suffered from the appearance of the so-called "disconnected" pixels [6] occurring in areas not conforming to the rigid translational motion model (e.g., expansion, contraction, rotation) and in occluded/exposed areas.

Research efforts to overcome this intrinsic problem of block-based motion model [7] failed to produce efficient framework that would successfully compete with state-of-the-art hybrid DCT coders. Recently, deformable-mesh motion models have been proposed for DWT video coding [2]. 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, 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 existence of the aforementioned "disconnected" pixels. On the other hand, 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 transform used. It is important to note, that motion overhead in mesh-based coders can stay comparable to that of motion-compensated hybrid DCT coders.

In this paper, we propose a new mesh topology that improves motion compensation at frame boundaries. We also propose a modification of the sequential iterative motion estimation process that prevents potential outliers, likely to form at frame boundaries, from affecting estimates in areas closer to image center. We demonstrate that the modified mesh topology doesn't have a significant impact on motion overhead, while improving compression performance.

2. REGULAR-MESH MOTION ESTIMATION

Mesh-based motion models have been shown to be a good alternative to block-based models, such as those used in



(a) Block-based motion model (b) Standard triangular mesh (c) Proposed triangular mesh

Fig. 1. Examples of different node-point topologies.

block matching. In a regular-mesh case, a 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. This is unlike block matching where reference-frame partitioning is regular (square blocks, e.g., 16×16, shown in Fig. 1(a)) but each block is a allowed a single dispalcement thus modeling accurately only rigid-body translational motion.

Two main issues in mesh-based motion estimation are: mesh topology and node displacement estimation. 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)). This configuration introduces a slight increase in the number of motion vectors needed to represent the complete motion field as compared to a block-based representation. If original video frames consist of $M \times N$ blocks, then MN motion vectors are needed for motion representation in standard block-based approach while $(M+1)(N+1)$ vectors are necessary in a triangular mesh motion representation. For the typical block size of 16×16 pixels and standard video resolutions, the increase in uncompressed motion information is 5% (ITU.R-601 - 720×480 resolution), 11% (SIF resolution) and 21% (QCIF resolution).

As for node displacement estimation, one possibility is an independent estimation of node-point motion vectors, e.g., by maximizing correlation over a small neighborhood of a node. However, since individual nodes are treated independently, motion compensation between node-points is often inadequate. As an improvement, regularized search-based solutions to node-point motion estimation with triangular and/or quadrilateral regular meshes and spatial transformations have been proposed [8, 9]. In particular, Nakaya and Harashima [8] proposed an iterative hexagonal matching procedure where the motion vector at a node point is estimated by local minimization of the prediction error. In each iteration, the current node motion vector is perturbed in order to minimize local intensity matching error in six triangular patches formed by the current node and its six neighbors. 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 triangular patches. Toklu *et al.* [9] extended this method by employing a hierarchy of regular meshes such that motion estimation on a coarse mesh provides initialization for the next (finer) level of the mesh.

For completeness, we note that a number of advanced content-based mesh motion estimation algorithms exist (for an extensive review see [10]). However, given typical computational complexity of content-based mesh motion estimation and additional overhead needed for the transmission of mesh topology itself, this approach is not yet a good alternative to regular meshes in practical near-real-time compression applications.

3. MODIFIED-MESH MOTION ESTIMATION

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; a motion correspondence cannot be established. This occurs when features get occluded or newly exposed due to, for example, object motion in the scene being filmed. Unfortunately, this happens most of the time, and thus most of the time parts of images have undefined (underlying) motion. In practice, however, since motion information is needed for compression at every position in the image, a motion representation is sought even in the occluded/exposed areas; some error norm is defined and parameters of motion representation are sought that minimize

this error. This computed motion has nothing to do with the true (underlying) motion in this area (which is undefined), but can be used effectively for compression. In the context of mesh-based motion estimation, this is addressed by adapting the mesh topology to image content, however this results in increased computational complexity and motion-overhead rate problems discussed in Section 2.

A particular case of occluded and 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 discrepancies 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 is needed. Important practical constraints on this new motion compensation are: acceptable computational complexity and regular mesh topology that is well suited to compression applications. We propose a new triangular mesh topology that shifts 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)). The motivation for this new topology is that smaller patches allow more accurate motion modeling close to frame boundaries, particularly in the presence of global motion. Moreover, the proposed modification in mesh topology prevents outliers at motion field boundary (likely to occur in typical motion scenarios) to affect larger patches that are now away from the frame boundary.

Unreliable motion estimation at frame boundaries is also a motivation for modifying the existing sequential search-based node-point motion estimation algorithms. Since in the hexagonal refinement displacements of neighboring nodes affect each other, erroneous vectors at image boundary may adversely affect their neighbors a bit further away from the boundary. In order to minimize this effect, we propose to start the iterative process of hierarchical hexagonal refinement on triangular patches closer to image center, and slowly include more and more nodes closer to frame boundary in subsequent iterations. In this way, reliable node-point displacements are calculated earlier on and are less affected by errors at image boundary. In a way, the motion field is more effectively regularized at frame borders thus dealing more efficiently with occlusion-prone (at the boundary) cases like global camera zoom-in. Note that constraints imposed on motion vectors by standard block-matching algorithms, such as not allowing to point outside frame boundaries, can often result in dramatic motion vector outliers. This inhomogeneity of motion field can then have an impact on the size of compressed motion information thus reducing the overall coding performance.

With the modified mesh topology, the number of nodes needed for complete mesh representation is now $(M+2) \times (N+2) - 4$, which represents a minor increase in motion information to transmit as compared to the block-based and standard triangular mesh configurations. For example, at SIF resolution the increase in the total number of nodes is approximately 22% as compared to block-based motion and 10% in comparison with standard triangular mesh configuration. However, smoother motion fields with fewer outliers should compress better, thus reducing the negative impact of the increased number of nodes. For typical video bit-rates this amounts to less than 4% increase in the total bit-rate compared to block-based motion.

4. 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 $\pm$ 8 pixels per frame for *Mobile & calendar* sequence and $\pm$ 16 pixels per frame for *Football* sequence 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.

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. Our experience shows that using non-regularized block-matching algorithm as an initial estimator of coarse motion vector field can introduce unwanted outliers and necessitate more iterations for convergence to correct mesh. In some cases, unregularized initialization can even destroy mesh connectivity. The search range and motion precision were kept the same as in block-matching case. For motion estimation, in the case of modified mesh topology, we used the strategy proposed in Section 3. In the first iteration, we estimate motion only at node-points whose distance from the frame boundary is larger than four inter-node distances. We then include more and more nodes into the estimation process in subsequent iterations, reaching border-nodes only when the "interior" motion will have already been well estimated.

In Fig. 2 we show motion bit-rates for 100 frames of both *Mobile & calendar* and *Football* sequences. The motion is encoded losslessly using JPEG-LS directly on arrays of horizontal and vertical motion components. Average motion bit-rates for *Mobile & calendar* sequence are 134 kbps, 142 kbps and 151 kbps for block-matching, standard mesh and modified mesh methods, respectively. These present 6% and 13% increase in motion bit-rates of mesh-based techniques compared to standard block-matching. As expected, the motion overhead rate increases slower than the number of nodes, which is due to the smoothing effect that regularization has on the estimated node motion. In the case of *Football*, this effect is even more visible because of very

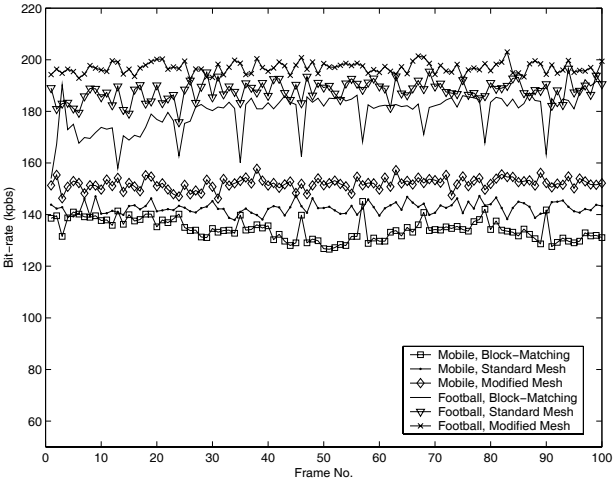


Fig. 2. Motion overhead bitrate

Table 1. PSNR performance [dB] at average of 500 kbps

Sequence	Block	Mesh	Modified mesh
Mobile & calendar	24.11	24.54	24.70
Football	22.78	24.11	24.37

dynamic motion in the sequence. Average motion bit-rates increase from 179 kbps for block-matching, to 187 kbps in the case of standard mesh (4% increase), and to 196 kbps (9% increas) in the modified mesh case.

Table 1 shows the PSNR performance for both sequences at the average rate of 500 kbps. We have applied JPEG2000 coder to each subband obtained with two decomposition levels of the motion-compensated (2,2) lifting transform. Note the slight PSNR gain of the modified mesh over regular mesh and a more significant one over block motion model, both gains despite increased number of motion parameters to transmit.

5. CONCLUSIONS

We have studied mesh-based motion models in the context of wavelet video coding. We proposed a new topology for triangular deformable meshes that improves motion estimation near frame boundaries while preserving mesh regularity. An increased number of node-points in the proposed scheme, however, doesn't have significant impact on motion bit-rate overhead due to improved regularity of the motion field. We achieve an additional improvement by starting the estimation process closer to frame center and including boundary node-points in subsequent iterations. This regularizes the motion field even for occlusion-prone (image boundary) motion types, such as global zoom-in. We have shown that invertible mesh outperforms standard block-based models and thus makes it a strong candidate for motion compensation in lifting-based DWT video coders.

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] 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).

[3] 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.

[4] 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.

[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 (to appear).

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

[7] S.-T. Hsiang and J.W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," in *Proc. Data Compression Conference*, 2001, pp. 83–92.

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

[9] C. Toklu, A. Erdem, M. Sezan, and A. Tekalp, "Tracking motion and intensity variations using hierarchical 2-d mesh modeling for synthetic object transfiguration," *Graphical Models and Image Processing*, vol. 58, pp. 553–573, Nov. 1996.

[10] Y. Altunbasak, R.M. Mersereau, and A.J. Patti, "A fast parametric motion estimation algorithm with illumination and lens distortion correction," *IEEE Trans. Image Process.*, vol. 12, no. 4, pp. 395–408, Apr. 2003.