HIERARCHICAL ANCHORING OF MOTION FIELDS FOR FULLY SCALABLE VIDEO CODING

Dominic Rüfenacht, Reji Mathew, and David Taubman

Interactive Visual Media Processing Lab (IVMP), School of EE&T, UNSW, Australia

ABSTRACT

Traditional video codecs anchor motion fields in the frame that is to be predicted, which is natural in a non-scalable context. In this paper, we propose a *hierarchical anchoring* of motion fields at reference frames, which allows to "reuse" them at finer temporal levels – a very desirable property for temporal scalability. The main challenge using this approach is that the motion fields need to be warped to the target frames, leading to disocclusions and motion folding in the warped motion fields. We show how to resolve motion folding ambiguities that occur in the vicinity of moving object boundaries by using breakpoint fields that have recently been proposed for the scalable coding of motion. During the motion field warping process, we obtain disocclusion and folding maps on-the-fly, which are used to control the temporal update step of the Haar wavelet. Results on synthetic data show that the proposed hierarchical anchoring scheme outperforms the traditional way of anchoring motion fields.

Index Terms— Hierarchical Motion Coding, Motion Warping, Breakpoint-Adaptive DWT, Fully Scalable Video Coding.

1. INTRODUCTION

Fully scalable video coding aims to exploit the redundancy between different quality levels of a video by encoding the video at the highest resolution, in a way such that partial streams can be decoded at lower resolution (spatial scalability), frame rate (temporal scalability), as well as a given signal-to-noise ratio (SNR scalability)[1]. Wavelet-based scalable video coding (WSVC) represents a promising approach. Secker and Taubman [2] propose the lifting-based invertible motion adaptive transform (LIMAT), which incorporates a deformable mesh model for the motion that is able to model expansion and contraction of moving objects. To the extent that the motion is correctly modelled, the transform is invertible. However, as mentioned in Adami *et al.* [3], multiresolution approaches are unable to represent information locality at moving object boundaries.

Recent research has shown promising results for WSVC, given that the motion in the video is continuous. Lalgudi *et al.* [4] adopt a similar approach to the LIMAT framework to compress volume rendered images. They determine the underlying geometric relationship between volume rendered images, which is then incorporated into the lifting steps of a temporal wavelet transform. Experimental results show superior compression performance compared to H.264/AVC. Garbas *et al.* [5] show similar promising results in the context of wavelet-based multiview coding. They apply a 4D wavelet transform (3D spatio-temporal, plus 1D for disparities), and observe that the temporal correlation characteristics between neighbouring views are almost identical. Similarly, over a small time instance, the view correlation is nearly constant. In both [4] and [5], the motion discontinuities are properly handled because of some intrinsic properties of the setup. This highlights the importance of motion discontinuities in order to achieve rate-distortion (R-D) results similar to state-of-the-art single layer codecs, and motivates the modelling of motion discontinuities for general scenes.

Mathew and Taubman [6] propose a scheme that incorporates a representation of discontinuities using *breakpoints*, which are determined in a R-D optimisation framework. The resulting breakpoint-adaptive DWT (BPA-DWT) uses breakpoints to avoid wavelet bases from crossing discontinuity boundaries. They apply the BPA-DWT on depth map data, which results in a reduction of the magnitude of subband samples in the vicinity of discontinuities. Importantly, the breakpoint representation itself is also fully scalable in resolution and precision.

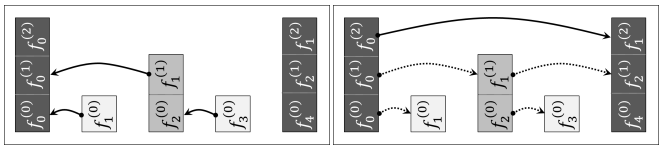
We argue that in order for fully scalable video coding to be competitive with traditional video coders, it would be beneficial to change the way motion estimation/compensation is performed by (1) using *optical flow fields* as opposed to *block motion fields*, and (2) anchoring motion fields at reference frames as opposed to target frames. The first change can be addressed by using *compression regularized optical flow* as advocated by Young *et al.* [7]. They jointly discover the motion field and breakpoints, which results in a piecewise smooth motion field that can be efficiently encoded using a BPA-DWT.

In this paper, we work with known motion fields, so as to focus on (2). We propose a new motion modelling scheme that is highly suited for fully scalable video coding. At this stage, our aim is not to propose a complete video codec. We highlight the advantages of the proposed *hierarchical anchoring* of motion fields in Sect. 2. In Sect. 3, we show that by using motion discontinuity information obtained from the breakpoint field, we are able to warp the motion field from the reference to the target frame and efficiently resolve folding ambiguities in the vicinity of motion discontinuities, by performing a 1D search. Sect. 4 shows how breakpoints can be temporally induced from coarse to fine temporal scales, which is particularly interesting at low bitrates. Experimental results are presented and discussed in Sect. 5, and Sect. 6 concludes the paper.

2. HIERARCHICAL MOTION FIELD ANCHORING

To simplify matters, in this paper we work only with the Haar temporal wavelet. At each temporal level, this means that the odd indexed frames are predicted using the preceding even indexed frame, while even indexed frames are updated using the prediction residuals.

A desirable property for temporal scalability is the ability to infer motion fields at finer temporal levels using motion at coarser levels. This is, at best, very difficult with the traditional anchoring of motion fields in the (odd indexed) prediction target frames, as employed in all current state-of-the-art codecs (*i.e.*, H.264/AVC, HEVC). Somewhat counter-intuitively, we propose to anchor the motion fields at the (even indexed) *reference frames*. Fig. 1 shows the traditional and our proposed hierarchical motion field anchoring schemes.



(a) Traditional anchoring (b) Hierarchical anchoring

Fig. 1. Different ways of anchoring motion fields: (a) Traditional anchoring at target frames and (b) The proposed *hierarchical anchoring* at reference frames.

We denote $f_k^{(t)}$ as frame k at temporal level t. In the traditional scheme, each predicted frame is assigned its own motion field. In the proposed hierarchical setup, however, only one motion field needs to be explicitly coded; the motion at finer temporal scales can be (partially) inferred from the coarser level (dotted arrows), so that finer temporal scales can be coded using differential coding. It is worth noting that the hierarchical anchoring allows motion to be scaled to even higher temporal frame rates than were originally available.

The traditional approach has been almost universally employed, primarily because it directly provides the motion vectors required for the temporal prediction step. As we shall argue, however, with the aid of breakpoints, we are able to infer a reliable motion field for prediction; in addition, our proposed *hierarchical anchoring* of motion fields provides many advantages for scalable coding.

2.1. Differential Coding of Motion Fields

In the following, we denote $M_{i\to j}^{(t)} (= M_{f_i^{(t)}\to f_j^{(t)}})$ as the motion field at temporal level t, anchored in frame i and pointing to frame j. As mentioned earlier, the hierarchical anchoring of motion fields facilitates predictive motion coding. At any given temporal prediction stage t, the forward motion field $M_{2k\to 2k+1}^{(t)}$ can be predicted from its parent motion field $M_{2k\to 2k+2}^{(t)}$ $(= M_{k\to k+1}^{(t+1)})$ by:

$$\hat{M}_{2k\to 2k+1}^{(t)} = \frac{1}{2} M_{2k\to 2k+2}^{(t)}. \tag{1}$$

Notionally, $M_{2k+1\to 2k+2}^{(t)}$ (*e.g.*, $M_{f_1^{(1)}\to f_2^{(1)}}$ in Fig. 1(b)) can also be inferred from $M_{2k\to 2k+1}^{(t)}$ and $M_{2k\to 2k+2}^{(t)}$ as:

$$\hat{M}_{2k+1\to 2k+2}^{(t)} = M_{2k\to 2k+2}^{(t)} \circ (M_{2k\to 2k+1}^{(t)})^{-1}. \tag{2}$$

Although none of the motion fields are likely to be truly invertible, the breakpoint dependent procedure that we propose for inferring these motion fields is well-defined and also possesses desirable smoothness attributes in regions where the true motion is uncertain.

Using these predictions, the motion fields can be differentially coded as:

$$\Delta_{M_{i\to j}^{(t)}} = M_{i\to j}^{(t)} - \hat{M}_{i\to j}^{(t)}. \tag{3}$$

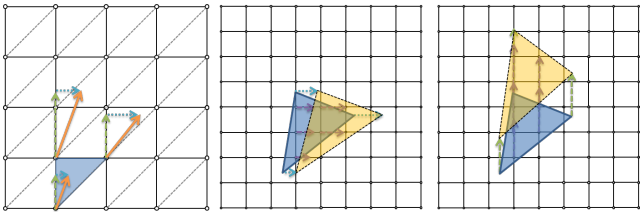
For compactness of notation, we omit the temporal level t in the following discussion.

3. WARPING OF MOTION FIELDS

The primary challenge of the proposed anchoring scheme is that of inferring a complete motion field $M_{2k+1\to 2k}$ in the target frame. Specifically, the challenge is to avoid *holes* and *double mapped regions* in the target frame because of *disocclusions* and *folding* of the motion field, respectively. In the following, we describe a procedure which naturally avoids holes, while exploiting breakpoints, where available, to disambiguate double mappings.

3.1. Cellular Affine Motion Warping

To generate $M_{2k+1\to 2k}$, we begin by partitioning the available motion field $M_{2k\to 2k+1}$ into small cells in the domain of the anchor frame f_{2k}. In this work we use 1×1 cells, dividing each cell into two triangles and assigning each triangle an affine flow based on the original motion,[1] as shown in Fig. 2.



(a) Reference (b) Hor. Motion (c) Vert. Motion

Fig. 2. Illustration of the cellular affine motion warping process.

Using this motion, each triangle is warped to the domain of the target frame, where covered samples are assigned the implied inverse motion. In order to mitigate aliasing in regions of local contraction, we work with an upsampled representation in the target frame.

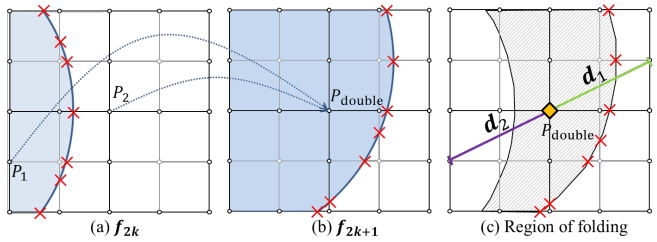
By suitably extending the reference frame motion field, we can guarantee that the warped triangles completely cover the target frame. This is because the warped triangles represent a continuously distorted affine mesh. In regions that are disoccluded in the target frame, the mesh is heavily stretched, which produces a smooth motion field in the target frame. The warped triangles can also overlap, so that multiple points in the reference frame map to the same point in the target frame; this corresponds to folding in the mesh. Both phenomena arise in the vicinity of discontinuities in the motion field. In the following, we show how to use breakpoints to resolve the ambiguities produced by folding.

3.2. Resolving Folding Ambiguities

Whenever a foreground object moves over background, there are parts of the background that are occluded in the target frame yet are visible in the reference frame and vice-versa. The first case leads to a *folding* of the motion field in the target frame, where a motion vector of the foreground moving object points to the same location as a vector from the background (moving) object. Clearly, only the motion from the foreground object is correct. The adopted motion coding framework uses breakpoints to efficiently code the motion itself, but here we use the breakpoints also to resolve ambiguities in the warped motion field. As described in [6] breakpoints describe the locations of motion discontinuity along the arcs (or edges) that run between grid-points at each level in a spatial hierarchy. In almost all cases, the breakpoints define line segments that coincide with motion boundaries. Fig. 3 shows a simple example with only translational motion.

A blue ball is moving two pixels to the right and one pixel up between f_{2k} and f_{2k+1}. For simplicity, we assume that the background is static. The grey shaded zone in Fig. 3(c) shows the locations that will be occluded in the target frame, which is where folding occurs. Let us focus on the orange diamond pixel. Let P_1 and P_2 be two points of the motion field of the reference frame which are mapped to the same point in the target frame, denoted here as P_{double}. Let $\vec{v} = P_2 - P_1$, which crosses a motion discontinuity in f_{2k}. We

[1] Clearly, the efficiency could be greatly improved by adopting larger cells in regions of smooth motion.

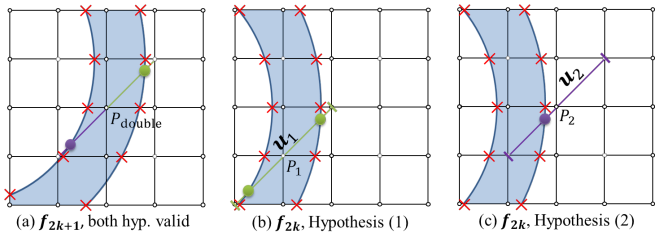


(a) f_{2k} (b) f_{2k+1} (c) Region of folding

Fig. 3. Resolving motion folding. Breakpoints are used to reason about the geometry of the scene.

have two hypotheses: (1) If the motion from P_1 is correct, then the line segment $\boldsymbol{d}_1 = [P_{\text{double}}, P_{\text{double}} + \vec{v}]$ should intersect with the motion discontinuity described by the breakpoints in the target frame f_{2k+1}, as the discontinuity "moves" with the foreground object. Similarly, (2) if P_2 is the correct foreground motion, the line segment $\boldsymbol{d}_2 = [P_{\text{double}}, P_{\text{double}} - \vec{v}]$ should intersect with the motion discontinuity in f_{2k+1}. In the present example, P_1 will be correctly identified as foreground motion.

In regions of high curvature in the discontinuities of the motion field, as well as with thin moving objects, it can happen that using the procedure described above, both directions lead to intersections with motion discontinuities described by breakpoints in the target frame f_{2k+1} (see Fig. 4(a)).



(a) f_{2k+1}, both hyp. valid (b) f_{2k}, Hypothesis (1) (c) f_{2k}, Hypothesis (2)

Fig. 4. For thin moving objects, both hypotheses are correct in the target frame. We therefore need to go back to the reference frame, where only the correct hypothesis will cross both motion boundaries.

In this case, we go back to the reference frame f_{2k} and look along the line segment $\boldsymbol{u}_i = [(P_i - |\vec{v}|), (P_i + |\vec{v}|)]$, for $i \in \{1, 2\}$. Because discontinuities displace with the foreground object, only one $\boldsymbol{u}_i$ will be found to intersect two motion boundaries in the reference frame; the associated i identifies P_i as the foreground motion.

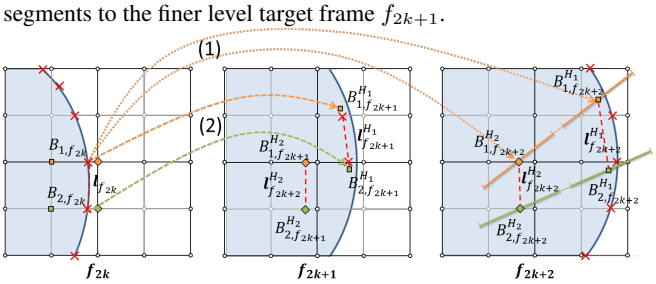
3.3. Disocclusion and Folding Map

The temporal warping of motion fields allows us to easily compute a disocclusion and folding map. This map is very useful for the temporal update step of the Haar wavelet, since regions that are labelled as *disoccluded* should not be used in the update as the prediction is unreliable. Since the results are less reliable in the vicinity of breakpoints in motion folded regions, the update step can be skipped at borders of folded areas.

4. TEMPORAL WARPING OF BREAKPOINTS

As described in the previous section, breakpoints play a key role in the proposed motion warping procedure. At lower bitrates, not all breakpoints might be available. In the following, we describe a method for warping breakpoints from the reference to the target frame, which allows for temporal induction of breaks that might have been scaled away. We want to warp each line segment described by two neighbouring breakpoints in f_{2k} ($\boldsymbol{l}_{f_{2k}} = [B_{1,f_{2k}}, B_{2,f_{2k}}]$) to frame f_{2k+1}. For this, we need to identify the motion that will warp $\boldsymbol{l}_{f_{2k}}$ closer to motion discontinuities described by breakpoints in f_{2k+2}; this will be the foreground motion.

The procedure we propose may be understood with the aid of Fig. 5, where a blue ball moves to the right and up in front of a static background (white). We proceed in two steps: (1) perform a *breakpoint compatibility check* (BCC) between the coarser temporal level frames f_{2k} and f_{2k+2}, which are expected to have the same precision in breakpoint information; and (2) warp *compatible* line segments to the finer level target frame f_{2k+1}.



f_{2k} f_{2k+1} f_{2k+2}

Fig. 5. Temporal Warping of breakpoints. (1) BCC, (2) Warping of compatible breakpoints.

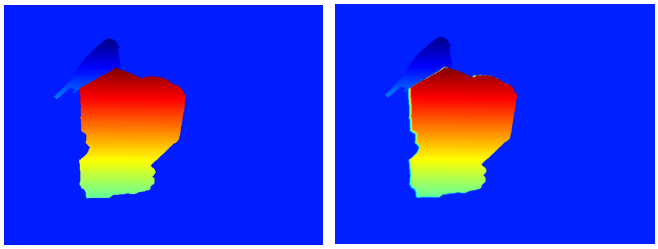
In step (1), we warp $\boldsymbol{l}_{f_{2k}}$ to f_{2k+2} under the hypotheses H_j ($j \in \{1, 2\}$) that the motion on side j is the foreground motion (in Fig. 7, side 1 is left, and side 2 is right). We denote the warped line segments as $\boldsymbol{l}^{H_j}_{f_{2k+2}}$. In order to determine which $\boldsymbol{l}^{H_j}_{f_{2k+2}}$ is closer to a motion discontinuity in f_{2k+2}, we create the *search* line segments $\boldsymbol{s}_i = [B^{H_1}_{i,f_{2k+2}}, B^{H_2}_{i,f_{2k+2}}]$, and extend them on both sides by half the length of $\boldsymbol{s}_i$. The breakpoint warped under H_j that is closer to the intersection of $\boldsymbol{s}_i$ with a line segment described by breakpoints (if any) in f_{2k+2} is marked as *compatible* . If there is a hypothesis H_j for which both ends of $\boldsymbol{l}^{H_j}_{f_{2k}}$ are compatible according to this test, then it is warped in step (2) to the line segment $\boldsymbol{l}^{H_j}_{f_{2k+1}}$ in f_{2k+1}, where all intersections with pixel grid lines are stored as inferred breaks.

5. EXPERIMENTAL RESULTS AND DISCUSSION

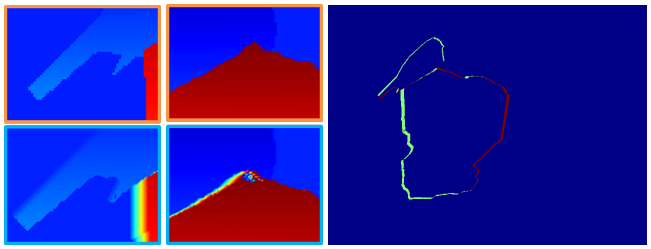
In the following, we present qualitative and quantitative results obtained on a synthetic sequence which contains two (partially overlapping) objects (a bird and a box) undergoing both rotational and translational motion on top of a static background (see Fig. 8 for an example frame). The bird is moving south and rotating counterclockwise, while the box is moving east and rotating clockwise. Therefore, motion folding arises east of the box, as well as south of the bird. We use ground truth motion data to illustrate the viability of the proposed method; the estimation of temporally consistent motion and breakpoint fields is part of ongoing research.

5.1. Qualitative Results

In this section, we show results for both the motion field warping as well as the breakpoint warping methods explained in Sect. 3 and Sect. 4, respectively. Fig. 6 shows an example of a warped motion field. We can see in Fig. 6(b) how the motion in disoccluded regions is stretched (west of box), and how the double mappings are correctly disambiguated (east of box). Fig. 6(c) shows how thin moving objects are handled, and Fig. 6(d) shows an example where there are too many motion discontinuities so that our algorithm cannot disambiguate the double mappings associated with motion folding. Fig. 6(e) shows the disocclusion and folding maps we obtain, which we use to guide the temporal update step. Fig. 7 shows an example of the breakpoint warping procedure described in Sect. 4. We

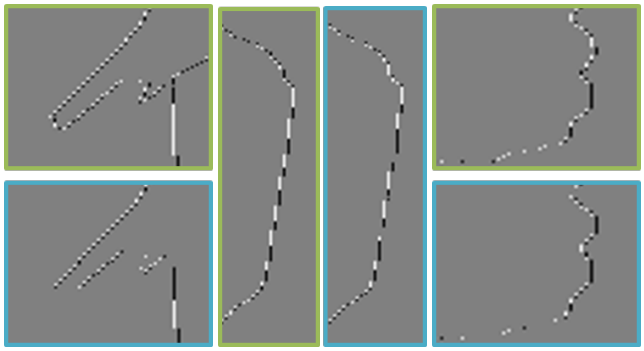


(a) Ground truth (b) Warped Motion Field



(c) Crop 1 (d) Crop 2 (e) DM and FM

Fig. 6. (a) Ground truth and (b) warped horizontal motion field. (c) and (d) are crops of (a) and (b). (e) is the obtained disocclusion map (DM) and folding map (FM) (green=DM, red=FM).



(a) Crop 1 (b) Crop 2 (c) Crop 3

Fig. 7. Details of a horizontal breakpoint map. Top/left are breakpoints computed on $M_{2k+1\to 2k+2}$, bottom/right are breakpoints warped from f_{2k} to f_{2k+1} (white/black = break right/left of pixel).

can observe that most breakpoints are found to be compatible, and mapped very close to the "correct" locations.

5.2. Quantitative Results

We present R-D graphs for various operating points of our proposed setup, as well as a comparison with the traditional way of anchoring motion fields. The sample data is compressed using three temporal levels, both for the proposed hierarchical as well as the traditional way of anchoring motion fields. The temporal subband frames are then subjected to a four level spatial DWT, followed by embedded block coding of the quantized wavelet coefficients. Motion fields are differentially coded as presented in Sect. 2.1. Motion and temporal subband frames are coded using JPEG2000, while breakpoints are coded using the method described in [6].

Fig. 8 shows R-D graphs of the average number of kbits used to encode four frames (which are needed for three temporal levels) for either using the differentially coded motion fields (DiffCoding=1), or just relying on the predicted motion (DiffCoding=0). For this preliminary work, we have not yet introduced progressive quality layers for the breakpoints. Instead, we report results for two cases: a) keeping all breakpoints in all frames (CodingBP=1); or b) discarding the coded breaks from all but the coarsest temporal level, from which the finer resolutions are inferred (CodingBP=0) using the breakpoint warping method of Sect. 4. The plot is the average bitrate for the 4 frames which are needed to encode three temporal levels employing a Haar temporal wavelet.

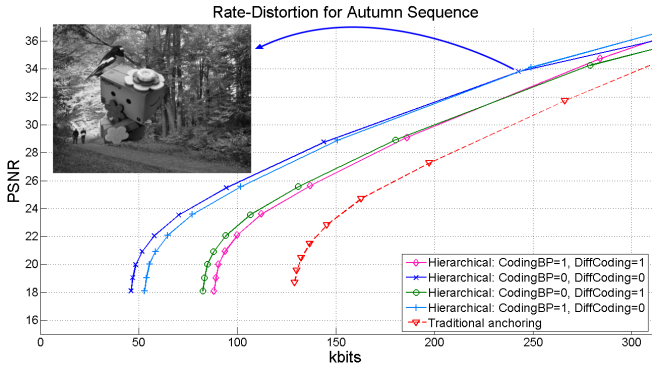


Fig. 8. R-D graphs for various configurations of the proposed hierarchical anchoring, as well as for the traditional anchoring.

We can see that the proposed *hierarchical anchoring* of motion fields outperforms the traditional way of anchoring for all configurations tested. This shows that the additional residual error in the texture data we get because of the warping of the motion field is smaller than the cost of coding the additional motion fields required in the traditional method. We also see that the temporal warping of breakpoints works well, resulting in lower bitrates for a given PSNR (CodingBP=0). Because there is little acceleration between the frames, the motion can be quite well predicted from motion at coarser temporal levels, and hence the performance is better when DiffCoding=0.

It is worth highlighting the fact that the proposed method allows highly credible reconstructed video even when all motion and breakpoint information is completely discarded from higher temporal levels(CodingBP=0, DiffCoding=0). Fig. 8 also shows a decoded frame obtained after discarding all such information.

6. CONCLUSIONS

We propose a new way of performing motion compensation that is better suited for fully scalable video coding. By abandoning the conventional wisdom that motion should be anchored at the motion compensated target frames, we find that scalability can be significantly enhanced. Specifically, by anchoring motion at the reference frames, we obtain a hierarchical representation that is more robust to the degradation (even absence) of finer motion scales. This approach requires that motion be warped from one frame to another. We have shown how this can be achieved with the aid of breakpoints that also play a key role in an efficient motion coding procedure that is itself fully scalable in resolution and quality. The proposed method implicitly handles holes (disocclusion) during the warping process, while motion folding is disambiguated using breakpoints without the need for complex geometric reasoning or full exploration of spatial neighbourhoods – at most a 1D search is required to resolve ambiguities. The proposed approach leads to motion fields that are piecewise smooth and consistent across time. Future work includes the extension to a temporal 5/3 wavelet transform, whose bidirectional motion attributes will improve the results in disoccluded regions.

7. REFERENCES

[1] J.-R. Ohm, "Advances in Scalable Video Coding," *Proceedings of the IEEE*, vol. 93, pp. 42–56, 2005.

[2] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression.," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–42, 2003.

[3] N. Adami, A. Signoroni, and R. Leonardi, "State-of-the-Art and Trends in Scalable Video Compression With Wavelet-Based Approaches," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1238–1255, 2007.

[4] H. G. Lalgudi, M. W. Marcellin, A. Bilgin, H. Oh, and M. S. Nadar, "View compensated compression of volume rendered images for remote visualization.," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1501–11, 2009.

[5] J.-U. Garbas, B. Pesquet-Popescu, and A. Kaup, "Methods and Tools for Wavelet-Based Scalable Multiview Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 2, pp. 113–126, 2011.

[6] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable coding of depth maps with R-D optimized embedding.," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1982–95, 2013.

[7] S. Young, R. Mathew, and D. Taubman, "Joint estimation of motion and arc breakpoints for scalable compression," *IEEE Global Conference on Signal and Information Processing (Global SIP)*, 2013.