Bidirectional, Occlusion-Aware Temporal Frame Interpolation in a Highly Scalable Video Setting

Dominic Rüfenacht, Reji Mathew, and David Taubman
Interactive Visual Media Processing Lab (IVMP)
School of Electrical Engineering and Telecommunications, UNSW, Sydney, Australia
{*d.ruefenacht, reji.mathew, d.taubman*}*@unsw.edu.au*

***Abstract*—We present a bidirectional, occlusion-aware temporal frame interpolation (BOA-TFI) scheme that builds upon our recently proposed highly scalable video coding scheme. Unlike previous TFI methods, our scheme attempts to put "correct" information in problematic regions around moving objects. From a "parent" motion field between two existing reference frames, we compose motion from both reference frames to the target frame. These motion fields, together with motion discontinuity information, are then warped to the target frame – a process during which we discover valuable information about disocclusions, which we then use to guide the bidirectional prediction of the interpolated frame. The scheme can be used in any state-of-the-art codec, but is most beneficial if used in conjunction with a highly scalable video coder. Evaluation of the method on synthetic data allows us to shine a light on problematic regions around moving object boundaries, which has not been the focus of previous frame interpolation methods. The proposed frame interpolation method yields credible results, and compares favourably to current state-of-the-art frame interpolation methods.**

I. Introduction

Temporal frame interpolation (TFI) consists of inserting frames at the decoder which are not present at the encoder. It is used in a variety of video coding applications, for example to reduce ghosting artefacts and motion blur in liquid crystal displays (LCDs) [1], or in distributed video coding, where temporally interpolated frames are used as side information for the Wyner-Zyv decoding [2]. In a *scalable video coding* context, where video can be decoded at different quality levels in terms of spatial, bit-rate, and temporal resolution, temporal frame interpolation is desirable when all information at a certain temporal level is quantized to zero.

We have recently proposed a *hierarchical motion field anchoring* for highly scalable video coding (HSVC) [3]. In this paper, we demonstrate how this anchoring can be adapted to obtain a bidirectional prediction framework with occlusion handling for TFI.

In current state-of-the-art codecs, motion fields are coded using blocks; each pixel in the *target* frame is assigned a vector pointing to the location in the reference frame where the block it belongs to matches best according to some error measure. This block motion does not in general represent the "true" motion, but one which minimizes the prediction error. It is therefore ill-suited to represent motion in the vicinity of motion discontinuities, and cannot be *scaled* to represent motion to intermediate frames. For these reasons, good-performing TFI methods first (re)estimate the motion between the two frames where a frame is to be inserted, which is then used to interpolate the target frame.

Wang *et al.* [4] perform motion-compensated prediction of the intermediate frame from both reference frames independently, and then blend these predictions together. Jeong *et al.* [5] perform motion-compensated frame interpolation using a multi-hypothesis motion estimation. The best motion hypothesis is selected by optimizing the cost function of a labelling problem. Pixels in the target frame are computed as a weighted combination of several pixels from the reference frame. Veselov and Gilmutdinov [6] propose a hierarchical bidirectional multi-stage motion estimation algorithm. They partition the target frame into non-overlapping, hierarchical blocks, and approximate the "true" motion flow. Each pixel is blended from multiple reference pixels. Chin and Tsai [7] estimate a dense motion field, and apply the motion to each pixel location. Simple heuristics are used to handle holes and multiple mapped locations in the upsampled frame.

The proposed TFI scheme differs from prior art in the following ways:

- Reliable *disocclusion* maps are computed, which guide the *bidirectional prediction* of the interpolated frame – we resort to appropriate unidirectional prediction in regions which are occluded in one reference frame;
- Estimated *motion field discontinuity* information allows to reliably identify the foreground object in regions of motion field folding (*i.e.,* resolve double mappings);
- If used with our HSVC scheme, motion fields which were estimated at the *encoding stage* can be (re)used, which reduces the computational complexity at the decoder.

In Sect. II, we show how our recently proposed scalable video coding scheme [3] can readily accomodate temporal frame interpolation. Sect. III shows how we estimate piecewise smooth motion fields suited for this work. We evaluate our method in Sect. IV using synthetically generated sequences. This is to get a better understanding of how problematic regions (disocclusions and double mappings) are handled. Another important benefit of using synthetic content is that it allows us to generate the most appropriate "ground truth" comparison frames – noting that all frame interpolation methods assume constant motion between the available reference frames, performance comparisons should be based on ground truth content that is consistent with this assumption.

II. Temporal Frame Interpolation in a Highly Scalable Video Coding Setting

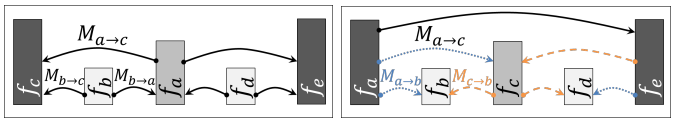
In our recently proposed highly scalable video coding scheme [3], we proposed to anchor motion fields at *reference* frames rather than *target* frames, which allows to reuse motion fields at finer temporal scales. In this section, we present how this scheme can be adapted to perform temporal frame interpolation. The proposed scheme uses piecewise smooth motion fields, together with a scalable encoding of discontinuities. The joint estimation of such motion fields and discontinuities is ongoing research; however, we describe a way of estimating motion fields that are suitable for this work in Sect. III.

For the remainder of this paper, we denote the two reference frames as f_a and f_c, respectively, and the frame to be interpolated (target) as f_b. We further use $M_{i\to j}$ to denote a motion field "anchored" at frame f_i and pointing to frame f_j, such that each pixel in frame f_i is associated with a location in frame f_j.

In the following, we present key parts of the proposed frame interpolation method: A) obtaining $M_{a\to b}$ as a *scaled* version of its parent $M_{a\to c}$, and *inferring* $M_{c\to b}$ from its "sibling" $M_{a\to b}$, and $M_{a\to c}$; B) inverting $M_{a\to b}$ and $M_{c\to b}$ to create motion fields anchored at f_b; C) transferring motion discontinuity information to f_b; and D) bidirectionally interpolating f_b using $M_{b\to a}$ and $M_{b\to c}$.

A. Bidirectional Hierarchical Anchoring of Motion Fields

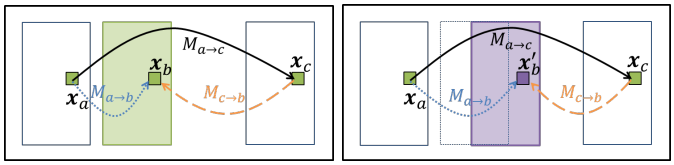
All current state-of-the-art codecs anchor motion fields at the *target* frames. In [3], we proposed to anchor motion fields at the reference frames instead. In this section, we demonstrate how the underlying methods of constructing motion fields are highly suited for frame interpolation, and can lead to a geometrically consistent bidirectional prediction of the target frame. Fig. 1 shows the two different ways of anchoring motion fields. Let us assume that all *odd* frames (f_b and f_d



(a) Traditional Anchoring (b) Hierarchical Anchoring

Fig. 1. a) Traditional anchoring of motion fields in the target frames, and b) *bidirectional hierarchical anchoring* of motion fields at reference frames.

in Fig. 1) are not present at the encoder, and we want to interpolate them at the decoder. In that case, $M_{a\to c}$ is the only motion field present at the decoder that can (potentially) be useful to interpolate frame f_b. In current state-of-the-art codecs, $M_{a\to c}$ is a block-based prediction field that minimizes the prediction residual, and is *not reflective* of "true motion". As a result, $M_{a\to c}$ cannot be *scaled* to point to the intermediate frame f_b, and hence has to be (re)estimated at the decoder. In our scalable video coding scheme, we closely model "true" motion fields, which can be *scaled* and hence readily be used to perform frame interpolation at the decoder. The remainder



(a) True motion (with acceleration) (b) Constant motion assumption

Fig. 2. A rectangle moves from left to right, with accelerated motion. a) shows the true location of the rectangle (green), and b) the predicted position of the rectangle under constant motion assumption. Note that because the *inferred* motion (orange dashed line) follows the *scaled* motion (blue dotted), the two motion fields $M_{a\to b}$ and $M_{c\to b}$ are geometrically consistent.

of the proposed method can be realized in traditional codecs; in that case, we also have to (re)estimate $M_{a\to c}$.

With a "true" motion field $M_{a\to c}$, one can readily compute a *scaled* version that points to the intermediate frame f_b, as $\hat{M}_{a\to b} = \alpha M_{a\to c}$ (typically $\alpha = 0.5$). In order to serve as prediction reference to interpolate frame f_b, we need to *invert* $\hat{M}_{a\to b}$. We present how motion fields are inverted for this work in Sect. II-B. Around moving object boundaries, there will be regions that get *disoccluded* (*e.g.,* uncovered) from frame f_a to f_b; such regions cannot be predicted from f_a. It is highly likely that such regions are visible in frame f_c, which is why we are interested in obtaining $M_{c\to b}$.

One could be tempted to estimate $M_{c\to a}$, and then compute $M_{c\to b}$ as a scaled version of $M_{c\to a}$. We defer from this strategy for two main reasons: 1) In a highly scalable video coder, this would be redundant information, and 2) it is highly likely that $M_{a\to c} \neq (M_{c\to a})^{-1}$, in particular around moving objects. Hence, their scaled versions will not be geometrically consistent in frame f_b. We instead *infer* $M_{c\to b}$ as follows: $\hat{M}_{c\to b} = M_{a\to b} \circ (M_{a\to c})^{-1}$. In order to be most useful, disoccluded regions in $\hat{M}_{c\to b}$ are filled with extrapolated background motion (see Sect. II-B). The fact that $M_{c\to b}$ is completely defined by $M_{a\to c}$ and $M_{a\to b}$ has the key advantage that $M_{c\to b}$ always "follows" $M_{a\to b}$, such that the two motion fields involved in the prediction of frame f_b are *geometrically consistent*. This highly desirable property is illustrated in Fig. 2.

B. Inversion of Motion Fields

We use the *cellular affine warping* (CAW) procedure first proposed in [9] to *invert* motion fields, which is guaranteed to leave no holes (in disoccluded regions); double mappings are resolved using *motion discontinuity* information to locally reason about foreground moving objects, as presented in [3].

During the composition of the *inferred* motion field $\hat{M}_{c\to b}$, we have to invert the "parent" motion field $M_{a\to c}$. Both $M_{a\to c}$ and $\hat{M}_{c\to b}$ should reflect "true" motion with sharp discontinuities. In particular, $\hat{M}_{c\to b}$ is most useful in regions which are not visible in frame f_a (*e.g.,* disoccluded). Under the assumption that the disoccluded regions belong to the background object, we use the *background extrapolation* technique explained in [3] to assign background motion to such regions in $\hat{M}_{c\to b}$.

The next step is to invert $\hat{M}_{a\to b}$ and $\hat{M}_{c\to b}$ to obtain $\hat{M}_{b\to a}$ and $\hat{M}_{b\to c}$, respectively. During this inversion process, we

use the "normal" affine motion model in disoccluded regions, noting that the affine motion leads to better results than extrapolated background motion in regions that are disoccluded in *both* reference frames (see Sect. II-D).

During the inversion of any $M_{i\to j}$, we readily observe disocclusions. We create a *disocclusion map* $S_{j\to i}$, where disoccluded locations $\mathbf{x}$ in $(M_{i\to j})^{-1}$ are assigned $S_{j\to i}(\mathbf{x}) = 0$, and all other *visible* regions as $S_{j\to i}(\mathbf{x}) = 1$. In the proposed bidirectional prediction setup, we obtain two such *disocclusion maps* anchored at the target frame f_b: one during the warping of $M_{a\to b}$, which we denote $S_{b\to a}$, and the other $S_{b\to c}$, obtained during the warping of $M_{c\to b}$, which are used to generate the interpolated frame as explained in Sect. II-D.

C. Hierarchical Warping of Motion Field Discontinuities

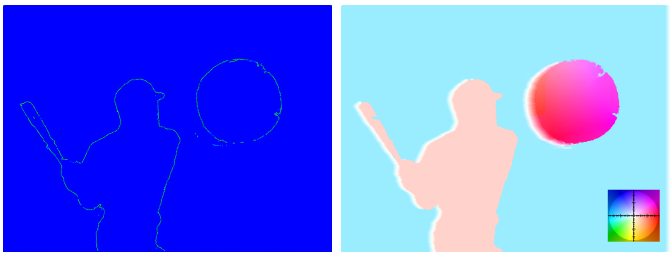
One key distinguishing feature of the proposed scheme is the use of motion discontinuity information to reason about scene geometry – it is used during the inversion of motion fields to resolve double mappings in regions of motion field folding, as well as to find the background motion used for the background extrapolation method employed during the creation of the *inferred* motion field $M_{c\to b}$. As this work builds upon a highly scalable video coding framework, we use a highly scalable way of coding discontinuities using *breakpoints* [8]. In essence, breakpoints lie on grid *arcs*, and can be connected to form discontinuity line segments.

For temporal frame interpolation, motion discontinuity information is not available for frame f_b. In this work, we transfer such discontinuity information from the reference frames to the target frame using a *hierarchical* extension of the breakpoint warping scheme proposed in [9], where discontinuity information was warped only at the finest spatial resolution. Observing that discontinuities "travel" with the foreground object, the algorithm consists in three main steps:

1) Breakpoint compatibility check (BCC) to find *compatible* (*i.e.,* foreground) motion to assign to discontinuity line segments;
2) Warping of *compatible* line segments under constant motion assumption to the target frame, where they are intersected with grid *arcs* and stored as breakpoints (*temporal induction*);
3) Upsampling of breakpoints to the next finer spatial resolution (*spatial induction*).

In cases where an intersected arc already contains a (spatially induced) breakpoint, the temporally induced breakpoint always overwrites the spatially induced one.

The advantage of this *hierarchical extension* is that the temporal inducing constraints are tightest at the finest spatial resolution; spatially induced discontinuity information from coarser spatial levels can help completing discontinuity information in regions that are not compatible at finer spatial resolutions, which is very desirable in the case of frame interpolation, where there is no residual information available to complete the breakpoint field. To sum up the two previous sections, we show a *temporally induced* breakpoint field of the target frame f_b, as well as a inverted motion field $\hat{M}_{b\to a}$ in Fig. 3 (obtained using estimated motion fields and breakpoints as presented in Sect. III).



(a) Warped breakpoints at f_b. (b) $\hat{M}_{b\to a}$

Fig. 3. Example of (a) warped breakpoints, which signal motion discontinuity information, and (b) a warped and inverted motion field used to predict f_b.

D. Frame Interpolation

The last step is to interpolate the target frame $\hat{f}_b$. We use $\mathcal{W}_{\hat{M}_{i\to j}}(f_j)$ to denote the warping process of frame f_j to frame f_i. The warping of frame f_j to frame f_i, evaluated at location $\mathbf{x}$, is then denoted as $f_{j\to i}(\mathbf{x}) = \Big(\mathcal{W}_{\hat{M}_{i\to j}}(f_j)\Big)(\mathbf{x})$. Every pixel location $\hat{f}_b(\mathbf{x})$ in f_b is computed using $\hat{M}_{b\to a}$ and $\hat{M}_{b\to c}$, together with the *estimated* disocclusion maps $S_{b\to a}$ and $S_{b\to c}$, as:

$$\hat{f}_b(\mathbf{x}) = \begin{cases} \dfrac{S_{b\to a}(\mathbf{x}) f_{a\to b}(\mathbf{x}) + S_{b\to c}(\mathbf{x}) f_{c\to b}(\mathbf{x})}{\kappa(\mathbf{x})} & \kappa(\mathbf{x}) > 0 \\ 0.5\big(f_{a\to b}(\mathbf{x}) + f_{c\to b}(\mathbf{x})\big) & \kappa(\mathbf{x}) = 0 \end{cases}, \tag{1}$$

where $\kappa(\mathbf{x}) = S_{b\to a}(\mathbf{x}) + S_{b\to c}(\mathbf{x})$.

Regions in f_b which are disoccluded in either of the reference frames (*i.e.*, $\kappa(\mathbf{x}) = 0$), are predicted from both reference frames equally, where the affine warping process guarantees that background texture is extrapolated.

III. Estimation of Piecewise Smooth Motion Fields with Discontinuities

In this section, we present how we estimate motion fields $\hat{M}_{a\to c}$ which are piecewise smooth with sharp transitions at boundaries of moving objects. Obtaining such motion fields in a rate-distortion optimal manner is a topic of ongoing research. The procedure we used for this work simply aims to show that such motion fields can be obtained, and that the proposed frame interpolation method works on estimated motion.

For the following discussion, we use index p to denote the frame *before* frame a in the sequence. We use Xu *et al.*'s [10] motion detail preserving optical flow algorithm to estimate the flow fields $M_{a\to c}$ and $M_{a\to p}$, where $M_{a\to p}$ is used to get sharp discontinuities in regions that are getting occluded between frames f_a and f_c. We start by assessing the quality of every motion vector of $\hat{M}_{a\to c}$ by motion compensating f_c, and computing the difference $\Delta_{f_a} = |f_a - \mathcal{W}_{M_{a\to c}}(f_c)|$. For each location where $\Delta_{f_a} < T$, where T is a fixed threshold, we consider that motion to be valid; otherwise, it gets labelled as *unassigned*. For all unassigned locations in the motion field, we assess the quality of the backward pointing motion field $M_{a\to p}$. Any valid motion will then be

inverted, and locally corrected for acceleration between frames f_p and f_c. The remaining unassigned pixels are assigned the estimated background motion, which for this work simply is defined as the largest connected component in the motion field. Discontinuities are then computed on $\hat{M}_{a\to c}$, using the breakpoint estimation scheme presented in [8].

IV. Experimental Evaluation and Discussion

In this section, we evaluate the proposed method both qualitatively and quantitatively on four synthetically generated sequences of varying degrees of complexity, which are shown in Fig. 4. *Space* is a sequence with differently accelerated

(a) Space (b) Baseball (c) Beach (d) Winter

Fig. 4. Frames of the different synthetic sequences generated.

objects, with reasonably complex geometry. *Baseball* has two foreground objects with large velocities and rotation, as well as accelerating background motion. *Beach* contains accelerated motion as well as rotations, with multiple objects intersecting. *Winter* contains a number of complex foreground objects (snowflakes) which intersect with another foreground object (the horse).

For our experiments, we generate for each sequence one version containing accelerated motion, whose frames are denoted f_k^{true}. We also generate another version of the same sequence, whose frames are denoted f_k^{const}, in which the motion between frames f_a and f_c follows the constant motion assumption that underpins all existing frame interpolation methods to our knowledge. This allows us to obtain the intermediate *target* frame f_b^{const}, which can be seen as the ground truth frame that should be interpolated under the constant motion assumption. We compare our results to two best-performing frame interpolation methods in the literature ([5] and [6]). All results and test sequences can be found on our website.[1]

A. Qualitative Evaluation

Fig. 5 shows qualitative results of the proposed temporal frame interpolation method on two of the test sequences. The heat maps of the prediction residuals show that all tested methods are able to predict regions within moving objects well; the main difference is how regions of occlusion are handled. One can appreciate how the proposed method is able to get much smaller errors in the vicinity of discontinuities, whereas the other methods smooth the texture. This is evidenced in the crops of the reconstructed frames, where in the *space* sequence, the satellite gets partially hidden between frames f_a and f_c, and hence can only be predicted from f_c. Also note how the proposed method is able to put the correct background information around moving objects, such as the tree branch to the upper right of the crab in row 4 (yellow rectangle).

[1] http://ivmp.unsw.edu.au/~dominicr/boa_tfi_results.html

TABLE I

Quantitative Comparison of the proposed method with [5] and [6]. We report both the PSNR of the reconstructed frame, as well as the PSNR in regions that are disoccluded in either of the two reference frames (occPSNR).

Sequence	Measure	Prop	Jeong [5]	Veselov [6]	Prop GT
Baseball	PSNR	**28.21**	27.15	25.61	31.67
	occPSNR	**23.08**	21.61	19.70	26.91
Beach	PSNR	31.09	**31.96**	29.23	34.00
	occPSNR	**23.86**	21.80	20.15	26.57
Space	PSNR	**29.47**	28.34	28.67	30.52
	occPSNR	**25.23**	22.51	21.79	26.27
Winter	PSNR	**24.33**	23.65	21.09	26.41
	occPSNR	**19.76**	17.50	15.33	22.15
Average	PSNR	**28.27**	27.77	26.15	30.65
	occPSNR	**22.98**	20.86	19.24	25.47

B. Quantitative Results

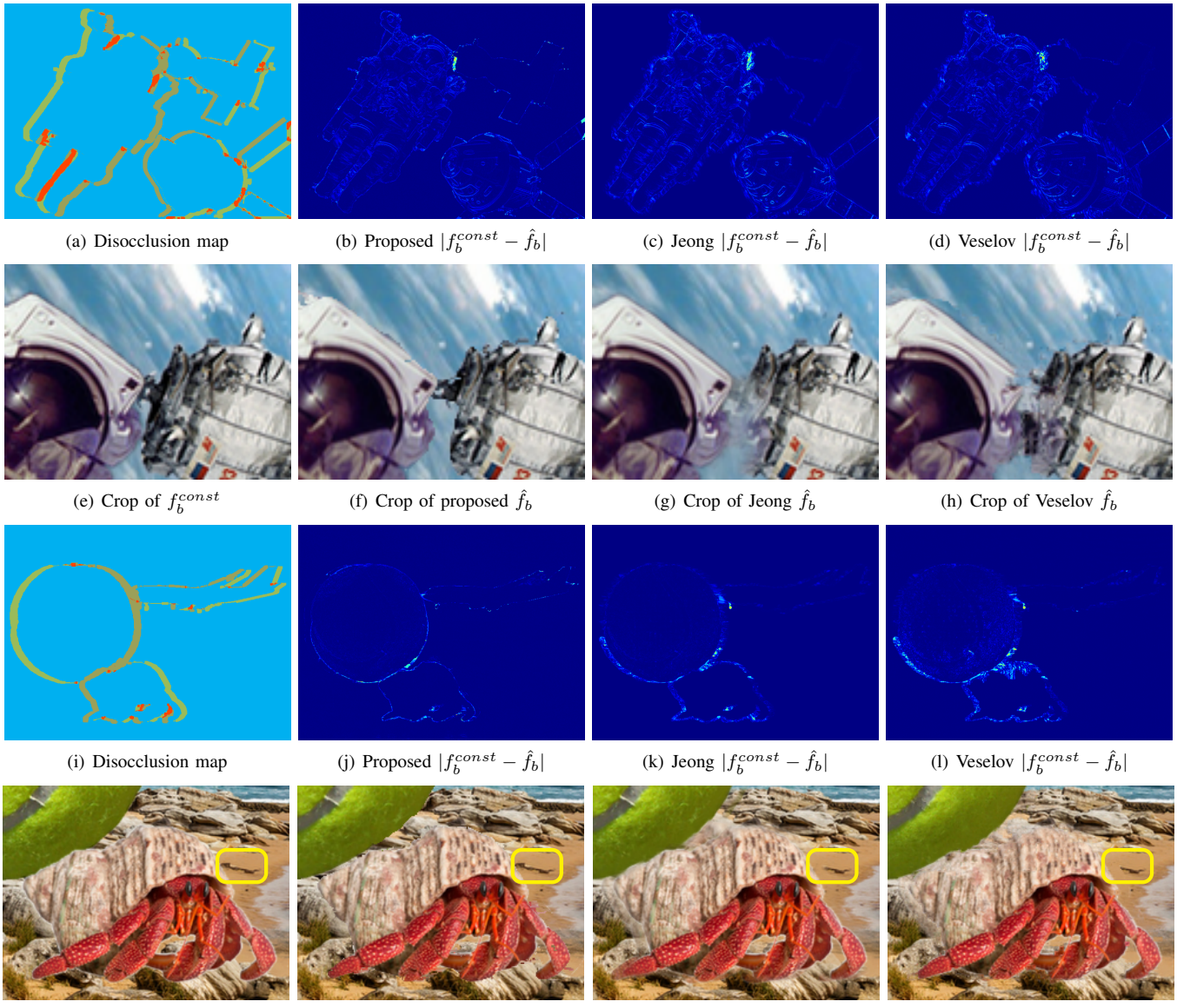
Table I presents quantitative results in terms of overall PSNR, as well as PSNR in occluded regions only (occPSNR). The table compares our proposed method with estimated motion (Prop) with those presented in [6] and [5]. The last column (Prop GT) reports the results of our method if we use the ground truth motion fields, which can be seen as an upper limit of what the proposed method is able to produce if the motion estimation step is improved.

With the exception of the beach sequence, where Jeong *et al.* have a lower prediction error around the plane, we have the best overall performance. A particular focus of this work is to better handle disoccluded regions in a more correct way. This is evidenced by the occPSNR value, where we outperform current state-of-the-art by around 2dB (using estimated motion) on the four tested sequences.

V. Conclusions and Future Work

This paper presents a novel way of performing temporal frame interpolation, which explicitly handles traditionally problematic regions around moving object boundaries. Using information about motion discontinuities, we are able to resolve double mapped regions in the target frame, as well as switching to an appropriate uni-directional prediction in regions that are occluded from one side. Evaluation of the method on synthetically generated video sequences shows that the method compares favourably to current state-of-the-art methods. Most noticeably, our method with estimated motion has almost a 2dB improvement in disoccluded regions, and over 4.5dB using ground truth motion fields. While the estimation and interpolation steps can be applied directly to the output from any current video codec, the proposed approach is especially beneficial if used in conjunction with a highly scalable video coder that employs the motion and breakpoint fields directly. In this case, the proposed method can be understood as an extension of the decoding algorithm, avoiding the need for (re)estimation of motion.

Ongoing and future work includes the joint estimation of piecewise-smooth motion fields and breakpoints for the scalable coding objective, as well as estimating and simulating optical blur around moving objects to reduce the "cut-out" effect our method exhibits around moving object boundaries.



(a) Disocclusion map (b) Proposed $|f_b^{const} - \hat{f}_b|$ (c) Jeong $|f_b^{const} - \hat{f}_b|$ (d) Veselov $|f_b^{const} - \hat{f}_b|$

(e) Crop of f_b^{const} (f) Crop of proposed $\hat{f}_b$ (g) Crop of Jeong $\hat{f}_b$ (h) Crop of Veselov $\hat{f}_b$

(i) Disocclusion map (j) Proposed $|f_b^{const} - \hat{f}_b|$ (k) Jeong $|f_b^{const} - \hat{f}_b|$ (l) Veselov $|f_b^{const} - \hat{f}_b|$

(m) Crop of f_b^{const} (n) Crop of proposed $\hat{f}_b$ (o) Crop of Jeong $\hat{f}_b$ (p) Crop of Veselov $\hat{f}_b$

Fig. 5. Temporal frame interpolation results for two sequences. In rows 1 and 3, (a) and (i) show the union of the disocclusion maps from the forward and backward prediction, where light green is occluded in the f_a, dark green is occluded in f_c, and red are the regions that are occluded in both f_a and f_c; (b-d) and (j-l) we show heatmaps of the absolute difference images between the frame obtained using constant motion (f_b^{const}) and the interpolated frame $\hat{f}_b$, $|f_b^{const} - \hat{f}_b|$. Rows 2 and 4 show crops of the (upsampled) frame of the ground truth f_b^{const}, as well as the interpolated frames $\hat{f}_b$ obtained using our proposed method, Jeong *et al.* [5], and Veselov *et al.* [6], respecively.

REFERENCES

[1] S. H. Chan and T. Q. Nguyen, "LCD Motion Blur: Modeling, Analysis, and Algorithm," *IEEE transactions on Image Processing*, vol. 20, no. 8, pp. 2352–2365, 2011.

[2] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-monedero, "Distributed Video Coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[3] D. Rüfenacht, R. Mathew, and D. Taubman, "Bidirectional Hierarchical Anchoring of Motion Fields for Scalable Video Coding," *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2014.

[4] C. Wang, L. Zhang, Y. He, and Y.-p. Tan, "Frame Rate Up-Conversion Using Trilateral Filtering," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 886–893, 2010.

[5] S.-G. Jeong, C. Lee, and C.-S. Kim, "Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4497–4509, 2013.

[6] A. Veselov and M. Gilmutdinov, "Iterative Hierarchical True Motion Estimation for Temporal Frame Interpolation," *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2014.

[7] Y. Chin and C.-J. Tsai, "Dense true motion field compensation for video coding," *IEEE International Conference on Image Processing (ICIP)*, pp. 1958–1961, 2013.

[8] 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.

[9] D. Rüfenacht, R. Mathew, and D. Taubman, "Hierarchical Anchoring of Motion Fields for Fully Scalable Video Coding," *IEEE International Conference on Image Processing (ICIP)*, 2014.

[10] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1744–1757, 2012.