Motion Blur Modelling for Hierarchically Anchored Motion with Discontinuities

Dominic Rüfenacht[1], Reji Mathew[2], and David Taubman[3]

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

[1] d.ruefenacht@unsw.edu.au
[2] reji.mathew@unsw.edu.au
[3] d.taubman@unsw.edu.au

***Abstract*—We have previously proposed a scheme for representing motion with motion discontinuities which has beneficial properties in terms of compactness (efficiency) and scalability. This so-called BIHA scheme has applications in video coding as well as temporal frame interpolation. In both cases, modelling of motion discontinuities has proven to be valuable. In these earlier works, we have ignored effects of motion blur, which can result in artificial sharp transitions of texture information at moving object boundaries. In this paper, we extend the BIHA framework to account for motion blur. Experimental results show significant improvements over the original BIHA scheme in texture blending regions, resulting in more visually pleasing predictions, as well as better rate-distortion performance.**

I. Introduction

This paper is part of our ongoing exploration of a novel motion modelling scheme for wavelet-based scalable video coding. Traditionally, motion fields are anchored at target frames. Recently, we proposed a scheme where motion is described in *reference* frames, and mapped onto target frames using a motion-discontinuity aware cellular affine warping process; we coined the scheme with the term *bidirectional hierarchical anchoring of motion fields* (BIHA) for video coding [1]. During the motion mapping process, reasoning about motion discontinuities allow to discover important properties the motion is undergoing (e.g., disocclusion and folding), which are beneficial to guide the bidirectional prediction of the target frames.

One drawback of the methods we have proposed so far is that they create overly sharp transitions in the motion-compensated texture information around moving object boundaries. In such regions, foreground and background texture information is usually blended together because of optical blur, and – more dominantly – *motion blur*. This paper addresses this problem. Motion blur occurs because each pixelsite of a camera sensor integrates incoming light over the exposure time, which blurs the image of moving objects. At moving object boundaries, a pixel integrates information from different objects, which results in a blending of foreground and background texture information; we refer to such regions as *texture blending* (TB) regions.

Handling motion blur in video coding is quite different from fields such as computer vision and image restoration applications, where the aim is to deblur image data affected by blur; blind-deconvolution methods with spatially uniform blur [2] as well as non-uniform blur [3], [4] have been extensively studied, but remain extremely challenging. In video coding, the aim is to predict certain *target* frames from existing *reference* frames. Current state-of-the-art video codecs (HEVC [5]) are agnostic to motion blur; as long as the reference frame is equally blurred, the target frame can be well predicted. Laude *et al.* [6] attempt to handle motion blur in HEVC by creating (additional) blurred reference frames, which they claim are helpful in regions affected by non-uniform motion blur. Because of the opportunistic nature of predicting "motion" in HEVC, it is hard to incorporate any more advanced motion blur handling than they propose.

In our BIHA scheme, on the other hand, the use of piecewise-smooth motion fields which are much closer to the "true" underlying motion field, allows for more advanced handling of motion blur. Obtaining such motion fields is a hard problem and a topic of ongoing research [7], in particular in scenes that are affected by motion blur. Wulff and Black [8] use a layered motion model and show how to jointly estimate piecewise-smooth motion fields as well as object boundaries in sequences that are affected by substantial motion blur.

In a temporal frame interpolation (TFI) framework, where the task is to increase the framerate of a video by inserting frames in between existing frames, motion fields are inherently "anchored" at reference frames. Successful TFI schemes hence start by (re)estimating the motion between the two frames where a frame is to be inserted [9], [10]. In [11], we have shown that the BIHA scheme naturally lends itself to a TFI framework with occlusion handling; in particular, the frames to be interpolated can simply be treated as frames for which all residual texture, motion, and discontinuity refinement information has been quantized away to 0. In this earlier work, we have ignored the effect of motion blur. To our knowledge, there exists no TFI scheme that explicitly addresses the problem of motion blur.

With this in mind, the current paper can be considered as a *contribution* to motion modelling for both video compression and temporal frame interpolation. In this work, we improve the

2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Oct. 19-21, 2015, Xiamen, China.
978-1-4799-5896-2/14/$31.00

BIHA framework in the problematic TB regions by proposing a way of estimating such regions, and consequently synthesizing motion blur by blending foreground and background texture information. We assume that the exposure time is known. The scheme we propose further assumes constant motion between the frames, noting that in the BIHA framework, estimates of acceleration could be derived. We evaluate the method using synthetic sequences, which is both convenient and illuminating, as it allows to focus on problematic regions.

We present the model we use in this work in Sect. II. In Sect. III, we present the proposed motion blur synthesis scheme as an extension of the BIHA scheme; in particular, we show how a reliable motion blur map as well as blending weights can be computed. Experimental results in Sect. IV show significant improvements over the BIHA scheme on sequences affected by motion blur.

II. A Generative Model for Sequences Affected by Motion Blur

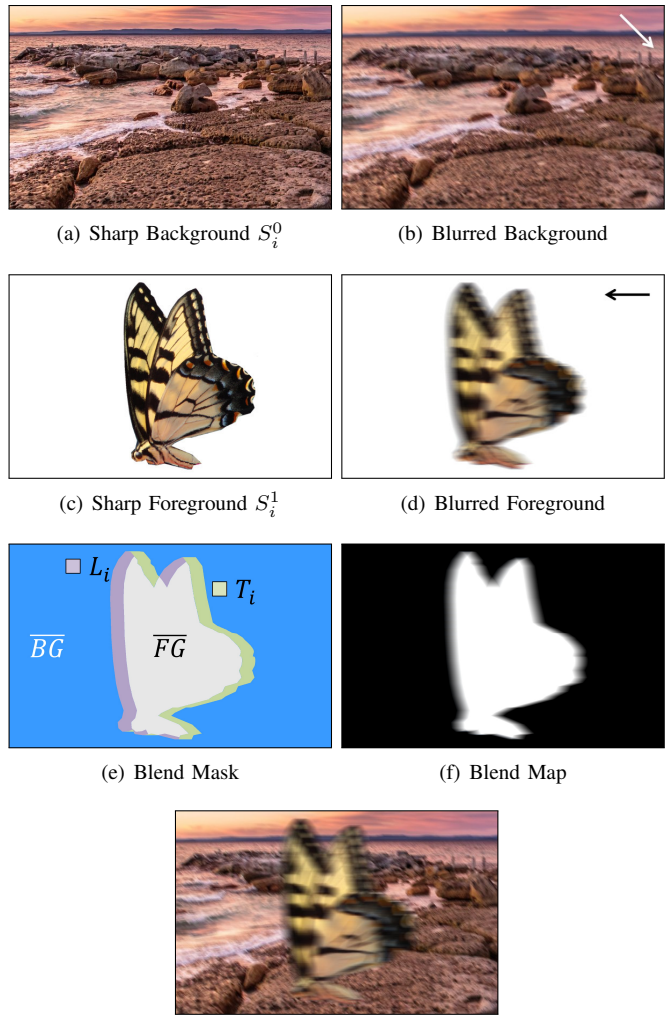
We start by presenting the generative model we use in the proposed framework, and introduce notation used throughout the rest of the paper. We use Fig. 1 to explain our generative model of a scene.

We assume that a scene is composed of multiple objects, all of which are possibly in motion. For the purpose of describing the model, let S_i^u denote a segmentation mask for object u, captured at time instance i; larger u means closer to the camera (i.e., $u = 0$ is the background). It is important to note that the proposed scheme does *not involve segmentation* in any way. In the real world, these objects are not affected by motion blur, and have a sharp appearance, as shown in Fig. 1a and c. We denote the frame that would have been captured with an infinitesimally short exposure time as f_i. In reality, the objects move during the exposure time τ (assumed to be known[1]), which results in blurred appearances (Fig. 1b and d). For the remainder of this paper, we call the hypothetical frame that would be captured at the end of the exposure time $f_{i'}$ $(= f_{i+\tau})$.

The foreground object moves on top of the background, and blends with the background texture information on both sides of the foreground object; these are the *texture blending* (TB) regions. We call the remaining regions *pure* foreground/background regions. For reasons that will become clear later, we use T_i^u (green) and L_i^u (magenta) to distinguish the trailing side and the leading side of the TB region of object u, respectively. More formally, $T_i^u = S_i^u \setminus S_{i'}^u$, and $L_i^u = S_{i'}^u \setminus S_i^u$. We further use $T_i = \bigcup_u T_i^u$ and $L_i = \bigcup_u L_i^u$ to denote the union of all trailing and leading blur regions, respectively. Fig. 1g shows the frame that is actually captured by the camera, which contains spatially varying blur. This frame is the combination of the blurred appearances of each object, weighted by the blending map (Fig. 1f).

Let f_p denote the next frame at temporal level t (i.e., $f_p = f_{i+\frac{2^t}{F}}$), where F denotes the frame rate. Note that in a real capturing scenario, $0 < \tau \leq \frac{1}{F}$. Assuming constant motion

[1]Such information can be communicated with negligible overhead.



(a) Sharp Background S_i^0 (b) Blurred Background

(c) Sharp Foreground S_i^1 (d) Blurred Foreground

(e) Blend Mask (f) Blend Map

(g) Captured frame f_i

Fig. 1. Generative model of a motion blurred sequence. (a) and (c) show the sharp foreground and background, respectively. (b) and (d) show the motion blurred objects. (e) shows a blend mask (green=trailing blur, magenta=leading blur, blue=pure foreground/background). (g) shows the frame captured by the camera, which contains spatially varying blur.

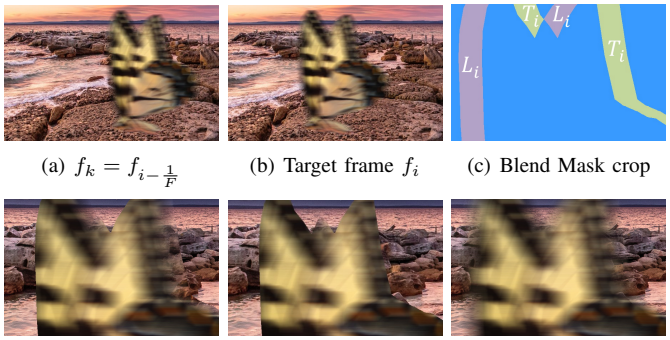
between the frames f_i and f_p, the motion vector which warps the frame f_i to $f_{i'}$ can be computed as a fraction of the motion vector between the two frames:

$$m_{i,i',\boldsymbol{x}} = \frac{\tau \cdot F}{2^t} \cdot m_{i,p,\boldsymbol{x}}. \tag{1}$$

The denominator scales the length of the motion blur trail back to the scale of the finest temporal level ($t = 0$). For the remainder of this paper, we assume that we are at the finest temporal level, and denote the next frame as f_j $(= f_i + \frac{1}{F})$.

In video coding, the task is to predict the blurred *target* frame from existing blurred *reference* frames. In the absence of acceleration, the only regions that do not have a corresponding location in at least one of the reference frames are the TB regions, where foreground and background information is blended together.

One might be tempted to ask why any coding effort should be spent on motion in TB regions; since such regions contain



(a) $f_k = f_{i-\frac{1}{F}}$ (b) Target frame f_i (c) Blend Mask crop

(d) BIHA (e) Shrunk FG $\tilde{f}_i$ (f) BIHA+TB

Fig. 2. Different stages of the proposed motion blur handling scheme. (a) shows one of the two reference frames involved in the prediction of the target frame f_i (b); (c) shows a crop of the blend mask at the target frame f_i, (d) shows the prediction of the BIHA scheme (note the wrong texture information in T_i, as well as the pure background prediction in L_i regions); (e) shows the intermediate prediction using the proposed modification of the BIHA scheme, which replaces T_i regions with pure background information; (f) shows the final prediction with synthesized texture blending in TB regions (BIHA+TB).

blurred texture, it may seem that a smooth interpolation between foreground and background motion should suffice in such regions. Adopting such an approach, however, effectively contracts texture information from the whole region that is disoccluded between f_i and f_j into the TB region, which creates sharp transitions in the texture; this leads to significant prediction errors, as evidenced in Sect. IV-B. We show in the following section how the BIHA scheme can be modified to properly handle motion blur.

III. Motion Blur Synthesis Framework

In this section, we present our method to accurately predict frames affected by motion blur. This work builds upon the BIHA framework [1], and uses *breakpoints* [12] as a highly scalable representation of *motion discontinuities*. The modified scheme is able to accurately model the blending of foreground and background texture in TB regions.

Since this framework naturally lends itself to temporal frame interpolation, the applicability of the proposed method is *not limited* to highly scalable video coding. In Sect. IV, we show results for both temporal frame interpolation as well as in a highly scalable video coding setting.

A. Method Overview

Fig. 2 shows two successive frames of a sequence where butterfly moves from right to left on top of static background; note that the scene is affected by motion blur. We start by giving a brief overview of the BIHA framework (for details see [1]). We use f_i to denote the target frame that is predicted from the previous and future reference frames $f_k = f_{i-\frac{1}{F}}$ and $f_j = f_{i+\frac{1}{F}}$. First, motion discontinuity information is transferred from the two reference frames to the target frame f_i. Next, we obtain $M_{k\to i}$ by *scaling* the parent motion field $M_{k\to j}$. $M_{j\to i}$ is then *inferred* from $M_{k\to j}$ and $M_{k\to i}$. Both $M_{k\to i}$ and $M_{j\to i}$ are *inverted* to obtain $M_{i\to k}$ and $M_{i\to j}$, respectively, which are used to bidirectionally predict the target frame f_i.

Fig. 2d shows the bidirectional prediction of the BIHA scheme. One can notice the sharp transition from foreground to background texture on the *leading* side of the moving object (L_i in Fig. 2c). This is because the motion fields are anchored at the *beginning* of the exposure time, and hence the lead blur region L_i will be assigned background motion. On the *trailing* side of the object in motion, there is also wrong information predicted in the texture blending region T_i. This comes from the fact that the motion assigned in this region is foreground motion. Therefore, the whole T_i region, where texture information gradually transitions from foreground to background, will be cut and pasted into the target frame. The problem is that the background moves with a different motion, and hence the background in the transition will be wrongly predicted.[2] Observing that there is one reference frame which contains the appropriate background information, we propose to predict an intermediate frame $\tilde{f}_i$ as shown in Fig. 2e, which contains pure background information in all TB regions.

The frame $\tilde{f}_i$ contains pure background information in all but the pure foreground regions, which results in a shrunk foreground object. The task is then to blend an appropriate amount of foreground texture in, which completes the proposed motion blur synthesis. Fig. 2f shows the result of the BIHA+TB scheme we propose in this work.

B. Estimating TB Regions

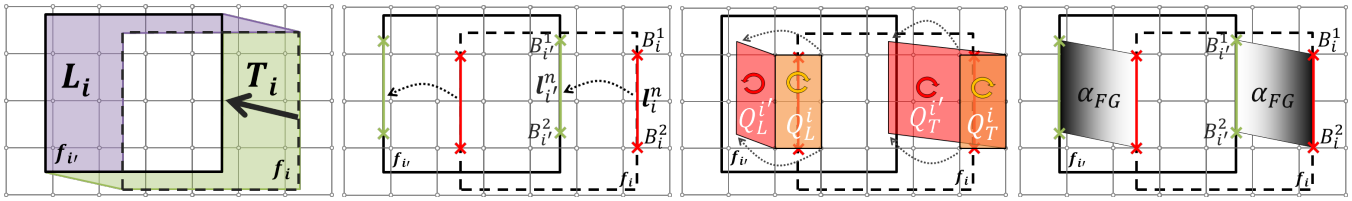
We now describe how TB regions can be estimated using information about motion discontinuities. For this, we augment the breakpoint warping procedure presented in [13]; in brief, the two main stages are: (1) assess the compatibility of breakpoint line segments between two reference frames by performing a *breakpoint compatibility check* (BCC); and (2) warp all compatible line segments to the target frame.

The core idea is that the breakpoint warping procedure allows us to warp discontinuity information from a frame to any arbitrary point in time by scaling the motion vectors appropriately. We hence are able to warp discontinuity line segments from f_i to the end of the exposure time ($f_{i'}$); the location of compatible line segments $\boldsymbol{l}_i^n$ in f_i are a good indication of the beginning of the motion blur region, while the line segment warped using the (identified) foreground motion, scaled according to Eq. 1 ($\boldsymbol{l}_{i'}^n$), marks the location of the objects at the end of the exposure time. Connected together, these line segments form a quadrilateral Q_{TB}^n; the estimate of the TB region is then simply the union of all these quadrilaterals, i.e. $TB = \bigcup_n Q_{TB}^n$.

We focus on one discontinuity line segment $\boldsymbol{l}_i^n$, formed by two breakpoints B_i^1, B_i^2 in frame f_i, and use Fig. 3 to guide the description.[3] We perform a BCC between f_i and f_j. All compatible line segments are then warped using the scaled foreground motion (see Eq. 1) to the (hypothetical) frame $f_{i'}$, which is at the end of the exposure time (Fig. 3b).

[2] The asymmetry of the artefacts in Fig. 2d comes from the fact that motion fields are anchored at the beginning of the exposure f_i.

[3] While the background is static in this example, the method is not restricted to static background (as exemplified by the "baseball" sequence in Sect. IV).



(a) A moving rectangle causes motion blur on the leading and trailing side. (b) Breakpoint Warping maps discontinuity line segments from f_i to $f_{i'}$. (c) Identifying L_i and T_i regions based on reasoning about orientation. (d) Blending weights map α_{FG} (white=1, black=0)

Fig. 3. Illustration how TB regions are identified and classified, as well as how the blending weights map α_{FG} is computed. (a) A rectangle moves from f_i (dashed) to f_i' (solid), on top of static[2] background; (b) Warping break line segments $\boldsymbol{l}_i^n$ from the beginning (f_i) to their corresponding location at the end of the exposure time ($f_{i'}$); (c) TB regions are classified based on whether the orientation (indicated by the "c-type" arrows) of Q_{TB}^i and $Q_{TB}^{i'}$ is the same or changes; (d) Assigning weights to the four breakpoints as detailed in Eq. 2 and then linearly interpolating between these points allows to compute a reliable foreground blending weights map α_{FG}.

C. Identifying Trailing and Leading Motion Blur Regions

Next, we identify whether a quadrilateral in the texture blending (TB) region (Q_{TB}^n) is on the trail or lead side of a moving foreground object, which is required to assign the correct blending weights in the next section. Consider the smallest quadrilateral that contains the line segment $B_i^1 - B_i^2$ in frame f_i, with vertices drawn from the sampling grid. Since B_i^1 and B_i^2 are breakpoints in the motion field at frame f_i, two vertices in this quadrilateral will have background motion while the other two will carry foreground motion. If the quadrilateral lies in the trailing side of the TB region (Q_T in Fig. 3c), the orientation of the mapped quadrilateral $Q_T^{i'}$ stays the same; quadrilaterals sitting in the leading side of the TB region (Q_L in Fig. 3c) and their mapped version $Q_L^{i'}$ have different orientation, as indicated by the "c-type" arrows.

D. Foreground and Background Texture Blending

The blending weights α_{FG} in the region occupied by the quadrilateral Q_{TB}^n (Fig. 3d) are obtained by assigning the following weights to its four vertices.

$$B_i^k = \begin{cases} 0 & Q_{TB}^n \in T_i \\ 1 & Q_{TB}^n \in L_i \end{cases}, B_{i'}^k = \begin{cases} 1 & Q_{TB}^n \in T_i \\ 0 & Q_{TB}^n \in L_i \end{cases}. \quad (2)$$

We then linearly interpolate between these endpoints to obtain the *foreground blending weights* for every quadrilateral Q_{TB}^n; combined together, these form a *foreground blending weights map* α_{FG} that contains the blending weights for foreground and background texture information in TB regions.

As mentioned in Sect. III, we modify the BIHA scheme to replace the foreground motion that is assigned to T_i regions with background motion; the obtained intermediate frame $\tilde{f}_i$ contains pure background texture information in the T_i regions. In order to complement the background texture information within TB regions, we record a motion vector $\boldsymbol{V}_{FG}(\boldsymbol{x})$ against each location $\boldsymbol{x}$ which points to representative foreground texture information. Using the blending mask, the final prediction of the target frame is computed by blending an estimate of the foreground texture with the background texture in TB regions.

$$\hat{f}_i(\boldsymbol{x}) = \alpha_{FG}(\boldsymbol{x})\tilde{f}_i(\boldsymbol{x}+\boldsymbol{V}_{FG}(\boldsymbol{x})) + (1-\alpha_{FG}(\boldsymbol{x}))\tilde{f}_i(\boldsymbol{x}). \quad (3)$$

Note that $\alpha_{FG} = 0$ in all locations $\boldsymbol{x} \notin \{T_i \cup L_i\}$.

IV. Experimental Results and Discussion

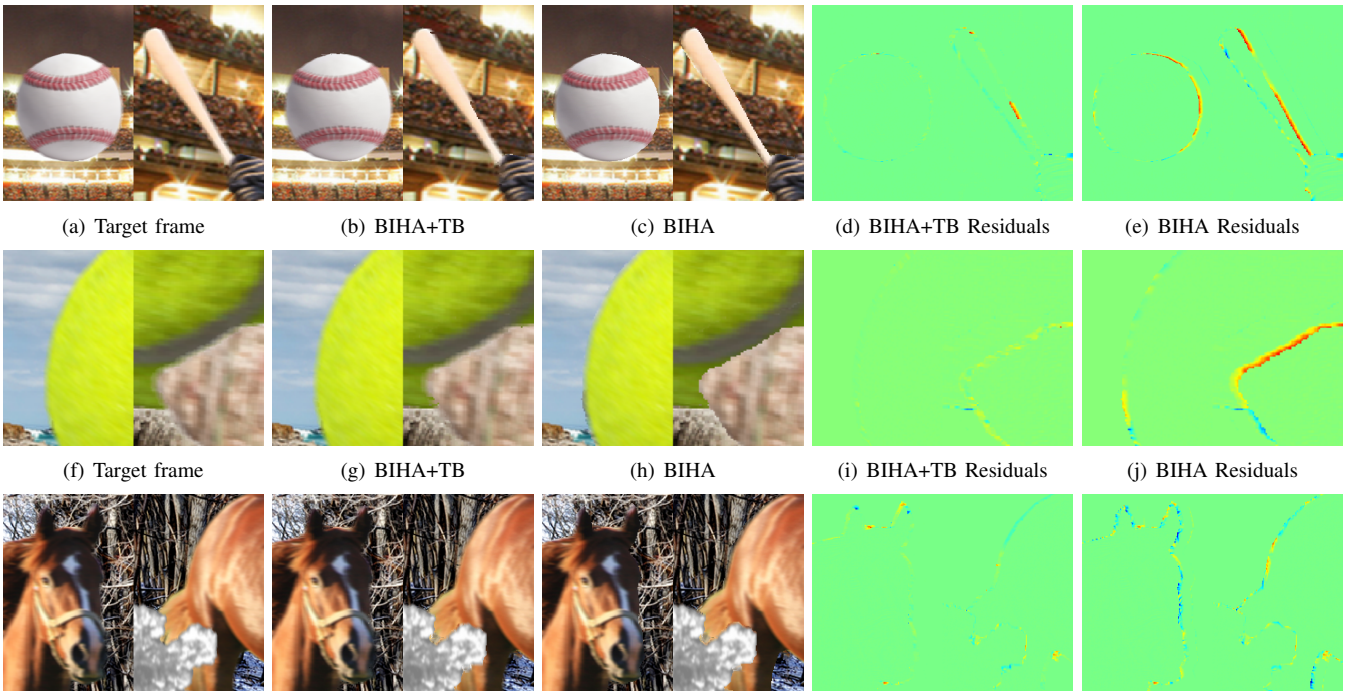
We show both qualitative and quantitative results for the proposed motion blur handling method. The proposed method employs motion fields that are piecewise smooth with sharp discontinuities. It is a topic of ongoing research to obtain such motion fields in the presence of motion blur. In this work, we generate synthetic sequences; the advantage is that we are in full control of the scene complexity, and have ground truth masks for regions of the texture blending (TB), which allows to fully investigate the benefits of the proposed motion blur handling method. It is important to note that *breakpoints* are *estimated* using the method presented in [12]. All sequences are available on our website,[4] along with all relevant ground truth motion fields and TB region masks. In the presence of motion blur, the location of objects is ill-defined, as they appear at different locations on the imaging sensor during the time of exposure. As mentioned earlier, we adopt the convention that motion fields are anchored at the beginning of the exposure time, which conforms with the motion fields estimated by Wulff and Black [8] on motion blurred sequences.

We start by evaluating the prediction capabilities of the proposed method by quantizing all residual information at the finest temporal level to 0, and keeping all other information at high quality; note that this operating mode can be seen as temporal frame interpolation.

A. Motion Blur Synthesis in Temporal Frame Interpolation

In this section, we show results from a temporal frame interpolation perspective, as this gives insight into how well the prediction scheme works. That is, given two reference frames $f_{i-\frac{1}{F}}$ and $f_{i+\frac{1}{F}}$, as well as a motion field $M_{i-\frac{1}{F}\to i+\frac{1}{F}}$, we want to interpolate a frame f_i in between these two reference frames. Fig. 4 shows example estimated blending maps, as well as the estimated frame, and Fig. 5 shows more qualitative results. One can see how the proposed BIHA+TB scheme has much lower prediction residuals in regions around moving objects, where foreground and background texture blend together. This shows how the proposed method is able to accurately predict TB regions, and is then able to blend the right amount of foreground and background texture together.

[4] http://ivmp.unsw.edu.au/~dominicr/biha_mblur.html

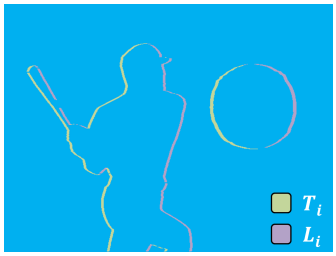


(a) Target frame (b) BIHA+TB (c) BIHA (d) BIHA+TB Residuals (e) BIHA Residuals

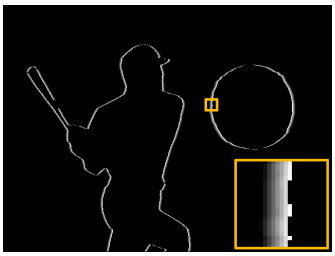
(f) Target frame (g) BIHA+TB (h) BIHA (i) BIHA+TB Residuals (j) BIHA Residuals

(k) Target frame (l) BIHA+TB (m) BIHA (n) BIHA+TB Residuals (o) BIHA Residuals

Fig. 5. Crops of temporally interpolated target frames (temporal frame interpolation). The first column shows the target frame; the second and fourth column show the predicted frame and residual, respectively, of the proposed BIHA+TB scheme with motion blur handling; Columns three and five show the prediction and residual of the BIHA scheme without motion blur handling, where large residuals can be observed in TB regions around moving objects.



(a) Blending mask



(b) Blending weights map α_{FG}

(c) Shrunk Foreground $\tilde{f}_i$

(d) Synthesized Motion blur $\hat{f}_i$

Fig. 4. Results of the proposed motion blur synthesis framework. (a) shows the computed blending mask, (b) the corresponding foreground blending weights (white=1, black=0), (c) are two crops of the predicted target frame prior to the foreground blending, and (d) final prediction of the target frame.

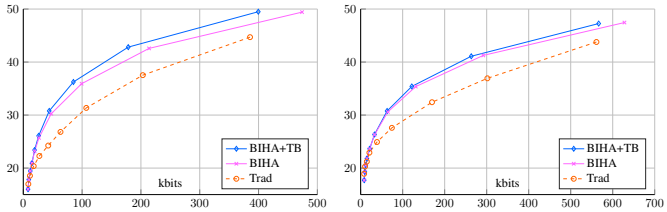
Because the scheme relies on breakpoints, regions where the breakpoint compatibility check fails are not properly handled, as can be seen in Fig. 5b/d, where part of the region around the baseball bat is assigned pure background texture. We are working on an improved discontinuity induction scheme that is expected to mitigate this problem. Table I shows quantitative results of the proposed motion blur synthesis scheme. We compare the proposed BIHA+TB scheme with the BIHA scheme without motion blur handling, and show both overall Y-PSNR, as well as Y-PSNR in texture blending regions (TB-PSNR); information about TB regions is obtained from ground truth data. One can see how the proposed motion blur synthesis leads to a significant improvement in terms of PSNR, which shows the effectiveness of the scheme. It is worth pointing out that the largest improvements are obtained for the baseball sequence; this sequence is probably the most complex one, with both background and foreground objects in motion.

TABLE I
COMPARISON OF THE BIHA SCHEME [1] AND THE PROPOSED BIHA+TB SCHEME IN TERMS OF PSNR AND TB-PSNR. THE LAST COLUMN SHOWS THE DIFFERENCE BETWEEN THE TWO SCHEMES.

Sequence	Measure	BIHA	BIHA+TB	Improvement
Baseball	PSNR	30.64dB	38.54dB	+7.90dB
	TB-PSNR	15.12dB	24.52dB	+9.40dB
Beach	PSNR	33.68dB	36.65dB	+2.97dB
	TB-PSNR	18.42dB	23.00dB	+4.58dB
Space	PSNR	34.78dB	37.32dB	+2.54dB
	TB-PSNR	20.87dB	24.28dB	+3.42dB
Winter	PSNR	28.36dB	32.30dB	+3.94dB
	TB-PSNR	14.95dB	20.29dB	+5.34dB

B. *Significance of Motion Discontinuities*

Many of the benefits of the proposed BIHA(+TB) framework come from the use of *motion discontinuities*. A natural



(a) R-D Baseball (b) R-D Space

Fig. 6. Average per-frame bit-rate against PSNR for various scenes, obtained using $T = 2$ temporal decomposition levels. We compare the proposed BIHA+TB method with our original BIHA scheme without blur handling, as well as motion fields anchored at reference frames with smooth motion in TB regions (Trad).

TABLE II
BD-PSNR AND BD-RATE GAINS OF THE PROPOSED BIHA+TB SCHEME COMPARED TO THE ORIGINAL BIHA SCHEME WITHOUT MOTION BLUR HANDLING, AS WELL AS A TRADITIONAL (TRAD) ANCHORING WITH SMOOTH MOTION IN TB REGIONS.

Sequence	BIHA		Trad	
	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate
Baseball	1.07dB	-12.05%	4.92dB	-45.37%
Beach	0.32dB	-4.58%	2.86dB	-32.36%
Space	0.34dB	-4.83%	3.12dB	-38.55%
Winter	0.45dB	-5.53%	5.92dB	-47.23%

question to ask is what happens if they are removed; this question is particularly relevant in the presence of motion blur. As a first step, we removed discontinuities from the BIHA framework, but the results were too poor to be reported. Instead, it is instructive to consider the *traditional* motion anchoring approach, where motion is anchored at the target frames. In that case, motion fields do not have to be inverted, and hence one can consider removing motion discontinuities to avoid the need to code them. Furthermore, we blurred motion in TB regions so that it smoothly transitions from foreground to background motion; we refer to this as the "Trad" approach.

The sample sequences are compressed using $T = 2$ levels of temporal decomposition. For the BIHA(+TB) scheme, the temporal subband frame textures, as well as the differentially coded motion fields, are then subjected to $D = 5$ levels of spatial, breakpoint-adaptive (BPA) DWT. Breakpoints are *estimated* coded using the method described in [12], and quantized based on the quality of the motion fields they are coding; intuitively, the more quantized the motion fields, the less breakpoints there are. The quantized wavelet coefficients are coded using EBCOT [14]. Fig. 6 shows rate-distortion curves for the three schemes, and Table. II shows BD-PSNR and BD-rate improvements of the proposed BIHA+TB scheme over the BIHA and Trad schemes.

The clear difference between the traditional anchoring (Trad) and the BIHA schemes indicates the value of signalling motion discontinuities. It is worth noting that the performance difference is not solely due to the lack of motion blur handling; less efficient motion field prediction and lack of identification of occluded regions also contribute to the poor performance. The results also confirm the benefits of motion blur handling in the BIHA framework, although the visual performance analysed in the previous section is perhaps more significant.

V. Conclusions and Future Work

We propose a method for synthesizing motion blur for a highly scalable video compression framework. The scheme employs piecewise-smooth motion fields together with *motion discontinuities*, which we show to be *useful* even if the object boundaries in the underlying frame texture data are blurred because of motion blur. We propose a method to compute reliable texture blending maps, which allow to significantly increase the prediction quality in texture blending regions. Qualitative and quantitative evaluation on synthetic sequences shows clear advantages of the proposed motion blur handling method, and motivates further research in realistic motion blur synthesis for video coding.

We assumed that the motion fields (and therefore motion discontinuities) are anchored at the beginning of the exposure time. We intend to relax this assumption and generalize the method to work with motion discontinuities anywhere in the TB region. We also like to include information about acceleration, which is encoded in our framework, in order to more correctly handle motion blur of accelerating objects.

References

[1] D. Rüfenacht, R. Mathew, and D. Taubman, "Bidirectional Hierarchical Anchoring of Motion Fields for Scalable Video Coding," *IEEE Int. Workshop on Multimedia Sig. Proc. (MMSP)*, 2014.

[2] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," *Proc. IEEE Conf. Comp. Vis. and Patt. Recognition (CVPR)*, pp. 1964–1971, 2009.

[3] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," *European Conf. on Comp. Vis. (ECCV)*, pp. 171–184, 2010.

[4] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," *IEEE Int. Conf. on Computational Photography (ICCP)*, 2011.

[5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding," *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] T. Laude, H. Meuel, Y. Liu, and J. Ostermann, "Motion blur compensation in scalable hevc hybrid video coding." *Pict. Coding Symp. (PCS)*, pp. 313–316, 2013.

[7] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, 2012.

[8] J. Wulff and M. J. Black, "Modeling Blurred Video with Layers," *European Conf. on Comp. Vis. (ECCV)*, vol. 8694, pp. 236–252, 2014.

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

[10] A. Veselov and M. Gilmutdinov, "Iterative Hierarchical True Motion Estimation for Temporal Frame Interpolation," *IEEE Int. Workshop on Multimedia Sig. Proc. (MMSP)*, 2014.

[11] D. Rüfenacht, R. Mathew, and D. Taubman, "Bidirectional, Occlusion-Aware Temporal Frame Interpolation in a Highly Scalable Video Setting," *Pict. Coding Symp. (PCS)*, May 2015.

[12] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable Coding of Depth Maps with R-D Optimized Embedding." *IEEE Trans. Image Proc.*, vol. 22, no. 5, pp. 1982–95, 2013.

[13] D. Rüfenacht, R. Mathew, and D. Taubman, "Hierarchical Anchoring of Motion Fields for Fully Scalable Video Coding," *Proc. IEEE Int. Conf. Image Proc.*, Oct. 2014.

[14] D. Taubman, "High Performance Scalable Image Compression with EBCOT." *IEEE Trans. Image Proc.*, vol. 9, no. 7, pp. 1158–70, Jan. 2000.