Leveraging Decoded HEVC Motion for Fast, High Quality Optical Flow Estimation

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

***Abstract*—We propose a method of improving the quality of decoded HEVC motion fields attached to B-frames, in order to make them more suitable for video analysis and enhancement tasks. We use decoded HEVC motion vectors as a sparse set of motion "seeds", which guide an edge-preserving affine interpolation of coded motion (HEVC-EPIC) in order to obtain a much more physical representation of the scene motion. We further propose HEVC-EPIC-BI, which adds a bidirectional motion completion step that leverages the fact that regions which are occluded in one direction are usually visible in the other. The use of decoded motion allows us to avoid the time-consuming estimation of "seeds". Experiments on a large variety of synthetic sequences show that compared to a state-of-the-art "seed-based" optical flow estimator, the computational complexity can be reduced by 80%, while incurring no increase at in average EPE at higher bit-rates, and a slight increase of 0.09 at low bit-rates.**

I. Introduction

All existing standardized video codecs, including the latest standardized video codec HEVC [1], exploit temporal redundancies between frames by partitioning so-called "target" frames into a set of disjoint blocks, each of which (potentially) gets one or multiple motion vectors assigned in order to drive a motion-compensated prediction (MCP) from already decoded reference frames. More precisely, for each block, weighted combinations of both the forward and backward block motion are used to form a prediction of the target frame block. This results in *opportunistic* block motion fields, which do not necessarily reflect "physical" scene motion between the frames. In particular, artificial motion discontinuities are introduced at block boundaries. Furthermore, blocks are unable to represent motion in the vicinity of moving object boundaries. Lastly, the opportunistic motion estimation employed in HEVC may result in no motion at all being communicated for some blocks, where spatial prediction is used instead.

For a variety of video analysis and enhancement tasks, a more physical representation of the underlying scene motion can be highly beneficial. For example, in the case of temporal frame interpolation (TFI), we have shown how such physical motion can be used to improve the quality of the interpolated frames compared to block-based TFI schemes [2]; furthermore, it opens the door to a meaningful incorporation of higher-order motion models [3], which can further improve the prediction quality.

978-1-5090-3649-3/17/$31.00 ©*2017 IEEE*

In computer vision, considerable progress has been made in the estimation of "optical flow" (OF) fields, both in increasing the quality of optical flow estimation [4], [5], [6], as well as reducing the computational complexity [7], [8], [9]. Nonetheless, high-quality OF methods are still quite far away from running in real-time on high-resolution content. A number of top-performing optical flow algorithms (e.g., [5], [8]), which we call "seed-based" OF, first estimate a sparse set of correspondences between the two frames where motion is to be estimated; these motion "seeds" are then interpolated using an edge-preserving interpolation strategy, called "EPIC" [8]. In the original EPIC method, the correspondences are found using deep matching [10], which accounts for the majority of the overall motion estimation time. In order to avoid the time-consuming estimation of correspondences, we proposed in [11] to use decoded motion vectors as "seeds" for the *edge-preserving interpolation* strategy (EPIC) [8]; we call the resulting method HEVC-EPIC (HE). These initial explorations were limited to *unidirectionally* predicted P-frames.

Motivated by these initial results, this paper continues our explorations of improving decoded HEVC motion for the widely used bidirectionally predicted frames. In particular,

- We extend HEVC-EPIC to work with *bidirectionally* predicted B-frames;
- We propose HEVC-EPIC-BI (Sect. III), which leverages bidirectional motion information to improve the quality of both the forward and backward motion fields.

To give some indication of the applicability of HE-BI motion for video enhancement tasks, we use HE-BI motion as input to a state-of-the-art temporal frame interpolation method [2], which was developed for the use of high quality motion fields. The quality of the interpolated frames using the proposed HE-BI motion is on par with what EPIC flow [8] is able to produce, while running over five times as fast.

II. Edge-Preserving Affine Interpolation of Block Motion Vectors

We start with a brief overview of the general idea behind the proposed HEVC-EPIC scheme, which aims at improving the motion field quality of decoded block motion fields; this overview is guided by Fig. 1. Following the assumption that motion boundaries are a subset of object boundaries in an image, we use the structured edge detector proposed in [12] to estimate an edge probability map of the target frame. We

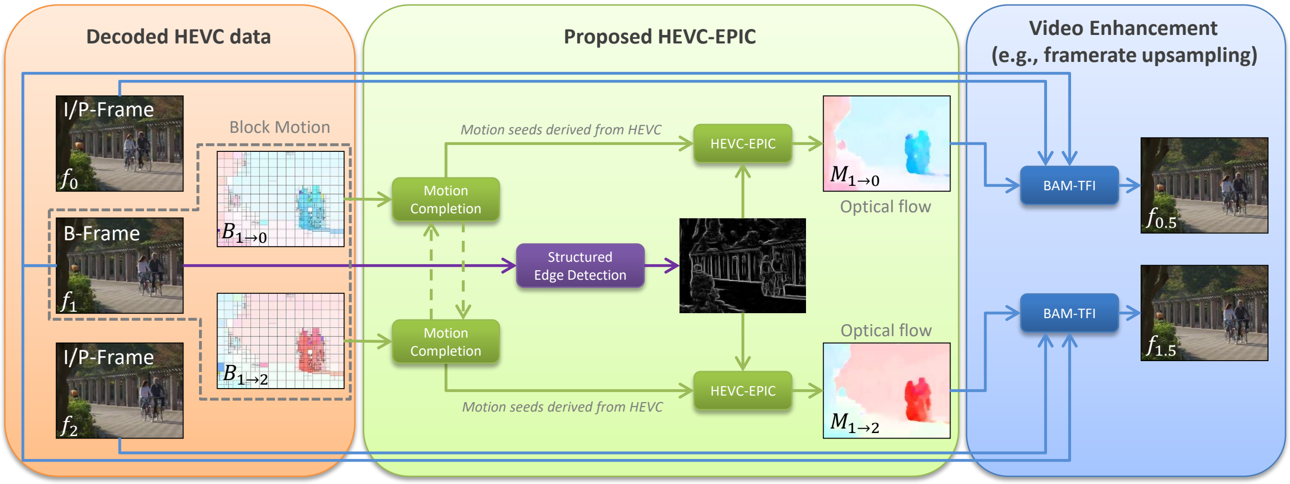


Fig. 1. Overview of the proposed HEVC-EPIC (HE) method. An input video sequence is coded using HEVC; in the example, we use an IBPB structure, and focus on a B-frame. For each B-frame f_k, block motion fields $B_{k\to k-1}$ and $B_{k\to k+1}$ are decoded. We estimate edge information on the decoded frame f_k using SED [12] (purple path), which is used to guide the sparse-to-dense affine motion interpolation procedure to obtain dense motion fields $M_{k\to k-1}$ and $M_{k\to k+1}$ that can then be used for video analysis and enhancement tasks (blue box), such as temporal frame interpolation (TFI).

consider the decoded motion vectors, anchored at the center of each block, as "seeds" for an edge-preserving affine interpolation method called EPIC [8]; the resulting dense motion field is devoid of artificial block boundaries, and contains sharp motion transitions around moving objects, which is a much more "physical" representation of the underlying scene motion than the decoded block motion.

In our initial proposal of HEVC-EPIC [11], we considered motion coded for *unidirectionally* predicted P-frames. For such frames, each block is either temporally predicted, which is called "Inter" prediction, or predicted using only spatial information of the target frame (so-called "Intra" prediction). In this work, we generalize [11] to *bidirectionally* predicted B-frames. For ease of explanation, we focus on an IBPB structure, where each B-frame f_k is predicted from its immediate temporal neighbours f_{k-1} and f_{k+1}; extensions to more general B-frame structures (e.g., hierarchical B-Frames [13]) can readily be obtained. In such a B-frame structure, each "Inter" predicted block is either bidirectionally predicted, or predicted only from either the previous (f_{k-1}) or future (f_{k+1}) reference frame.

Before we show how motion information from both motion fields can be leveraged in order to improve the quality of the proposed method in Sect. III, we give an overview of HEVC-EPIC method.

A. HEVC-EPIC (HE)

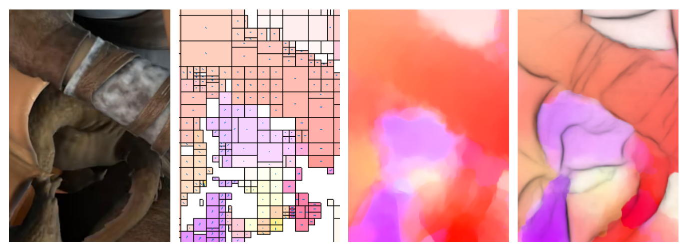
We now provide a description of how the decoded block motion vectors can be interpolated in order to obtain a dense motion field which preserves motion discontinuities. We use $B_{k\to j}$ to denote a decoded *block* motion field, which for each "Inter" block in the "target" frame f_k has a motion vector pointing to "reference" frames f_j, where we use $j = k-1$ or $j = k+1$ to refer to the previous or future reference frame, respectively. For the following discussion, let us use N_j to denote the number of motion vectors that are present in $B_{k\to j}$. As mentioned earlier, we propose to use these N_j motion vectors as "seeds" to drive an edge-preserving affine motion interpolation. For each motion vector $\mathbf{u}^n_{k\to j}$, $n \in \{1, ..., N_j\}$, we construct pixel correspondences $(\mathbf{x}^n_k, \mathbf{x}^n_j)$ as follows; here, $\mathbf{x}^n_k$ is the location of nth motion vector $\mathbf{u}^n_{k\to j}$ in frame f_k. Its corresponding location in frame f_j is

$$\mathbf{x}^n_j = \mathbf{x}^n_k + \mathbf{u}^n_{k\to j}. \quad (1)$$

In order to obtain a dense motion field from this sparse set of motion vectors, each location $\mathbf{m}$ of the *motion field* $\hat{M}_{k\to j}$ is interpolated using a locally-weighted *affine* estimator,

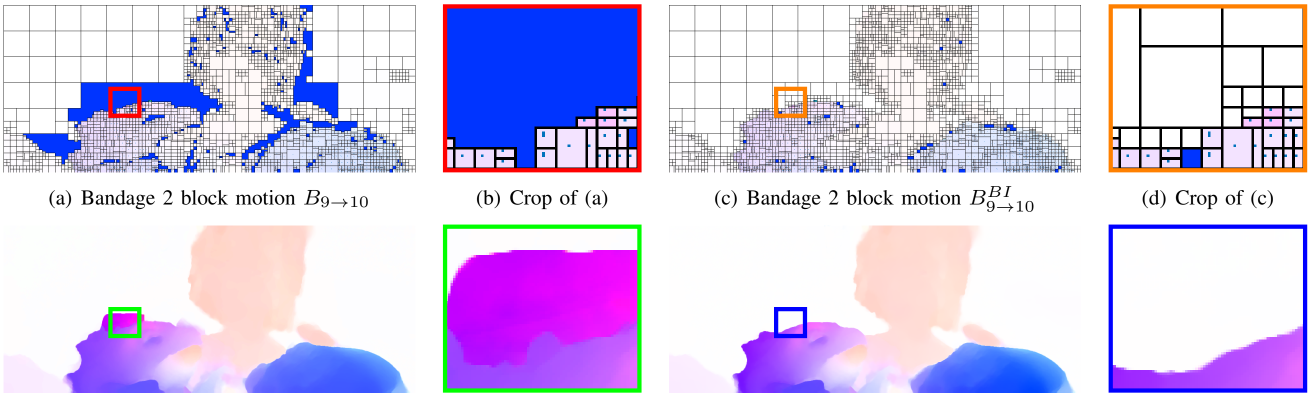
$$\hat{M}_{k\to j}[\mathbf{m}] = A_{\mathbf{m}}\mathbf{m} + t_{\mathbf{m}}, \quad (2)$$

where $A_{\mathbf{m}}$ (2×2 matrix accounting for rotation, zoom, shear) and $t_{\mathbf{m}}$ (2×1 vector accounting for translation) are parameters of the affine transform at pixel location $\mathbf{m}$. Note that at least *three* points $\mathbf{x}^s_i$ in frame f_i that are *closest* to $\mathbf{m}$, and



(a) Decoded f_k (b) $B_{k\to k+1}$ (c) Affine interp. (d) HEVC-EPIC

Fig. 2. (a) shows a decoded frame, and (b) shows a (colour-coded) forward referencing block motion field $B_{k\to k+1}$, where white regions are regions without motion assignment. (c) shows the affine interpolated motion of (b), which is smooth across moving object boundaries. (d) shows the result of the proposed HEVC-EPIC, where we superimposed the edge information (estimated using SED [12] on f_k) that was used to guide the edge-preserving affine interpolation.



(a) Bandage 2 block motion $B_{9\to 10}$ (b) Crop of (a) (c) Bandage 2 block motion $B^{BI}_{9\to 10}$ (d) Crop of (c)

(e) HEVC-EPIC $M_{9\to 10}$ (f) Crop of (e) (g) HEVC-EPIC-BI $M_{9\to 10}$ (h) Crop of (g)

Fig. 3. Bidirectional motion vector completion. (a) the decoded block motion $B_{9\to 10}$, where blue regions correspond to blocks without motion information. (b) the bidirectionally completed block motion field $B^{BI}_{9\to 10}$, where for each block where motion in $B_{9\to 8}$ is available, its negative value was assigned. (c) and (d) are the motion fields produced by HE(-B). (e) – (h) show crops of (a) – (d).

their corresponding locations $\mathbf{x}_j^s$ in frame f_j, are required. In order to increase the robustness to outliers, however, $S > 3$ correspondences can be used; in this work, we empirically set $S = 100$, and note that the method is not overly sensitive to the choice of S. Next, we compute the least-squares solution of the following overdetermined system

$$(A_{\mathbf{m}}, t_{\mathbf{m}}) = \underset{(A,t)}{\arg\min} \sum_{s=1}^{S} e^{-D(\mathbf{x}_k^s, \mathbf{m})} (A\mathbf{x}_k^s + t - \mathbf{x}_j^s), \qquad (3)$$

where $D(\mathbf{a}, \mathbf{b})$ is a *distance* measure between the points $\mathbf{a}$ and $\mathbf{b}$. Note how the affine model puts more importance on fitting points $\mathbf{x}_k^s$ that are "closer" to the location $\mathbf{m}$ of the point we seek to interpolate.

Next, we show the impact of the edge-aware distance measure. Fig. 2b shows a crop of a decoded HEVC block motion field, and Fig. 2c shows the corresponding affine interpolation; that is, each location $\mathbf{m}$ was interpolated according to (3), with Euclidean distance as distance measure $D(\cdot, \cdot)$. The resulting dense motion field is overly smooth; in particular, around motion discontinuities, foreground and background motion is averaged together, leading to non-physical motion. We use the edge-aware distance measure proposed by Revaud *et al.* [8], which imposes a large weight for paths that cross edges in the image. More precisely, the "cost" is measured by an edge probability map, estimated on the texture of the frame using a structured edge detector (SED) [12]. Fig. 2d shows the dense motion field obtained by applying (3) with the edge-aware distance measure, where we overlaid the edge probability map (black); one can see how the motion boundaries are much better preserved, resulting in a more "physical" representation of the underlying scene flow. In the following, we present a simple method of leveraging the bidirectional information to improve the input for HEVC-EPIC, which is particularly useful in occluded regions.

III. Bidirectional Motion Completion (HE-BI)

In *occluded* regions, which arise on the leading side of moving objects, no reliable motion can be estimated, as there will not be a correspondence in the corresponding reference frame. However, it is quite likely that such regions are visible in the other reference frame. In such regions, HEVC will presumably switch to unidirectional prediction. In Fig. 3a/b, all the blue regions correspond to regions where no forward motion is available; note how most of the blue blocks are around object boundaries, which are not visible in the future reference frame, and hence should not be predicted from that frame.

However, since we want to create a motion field as close to the ground truth as possible, we are interested in getting motion information in occluded regions as well. We therefore assign each block where no motion is available for $B_{k\to j}$ the negative of the motion in $B_{k\to p}$. That is,

$$\begin{aligned} B^{BI}_{k\to k+1}[n]|_{B_{k\to k+1}[n]=\emptyset} &= -B_{k\to k-1}[n], \text{ and} \\ B^{BI}_{k\to k-1}[n]|_{B_{k\to k-1}[n]=\emptyset} &= -B_{k\to k+1}[n]. \end{aligned} \qquad (4)$$

This can be seen as assuming a constant motion velocity in occluded regions. We refer to HEVC-EPIC that uses these "augmented" block motion vectors as HEVC-EPIC-BI (HE-BI). Fig. 3c/d shows an example of such a bidirectionally completed motion field. The impact on motion field quality when input to HEVC-EPIC can be appreciated in Fig. 3e/f, where around the dragon's head (cropped rectangle), foreground motion is "spilled" into the (occluded) background region in the original HE. In HE-BI, where the motion in the occluded region has been completed according to (4), this region is correctly interpolated, as evidenced in Fig. 3g/h.

IV. Experimental Validation

In this section, we evaluate the quality of the proposed HEVC-EPIC-BI (HE-BI) in an IBPB structure, with a QP offset of 2 for B-frames. We first provide a thorough experimental validation of the quality of the motion fields produced by HE-BI in Sect. IV-A. Motivated by these results, we show in Sect. IV-C the applicability of HE-BI motion for temporal frame interpolation.

TABLE I

AVERAGE EPE OF FORWARD MOTION FIELDS (A-EPE(R^{ALL})). WE COMPARE HE-BI (SEE SECT. III) TO ORIGINAL HEVC BLOCK MOTION, AS WELL AS WITH THE ORIGINAL EPIC FLOW [8]. WE ALSO SHOW OVERALL AVERAGES IN VISIBLE (A-EPE(R^{VIS})) AND IN ALL REGIONS WHERE $B_{k\to k+1}$ CONTAINS MOTION (A-EPE(R^{FWD})); HEVC RESULTS ARE GREY FOR UNFAIR COMPARISONS.

Sequence	QP=22 HEVC	QP=22 EPIC	QP=22 HE-BI	QP=27 HEVC	QP=27 EPIC	QP=27 HE-BI	QP=37 HEVC	QP=37 EPIC	QP=37 HE-BI
alley_1	0.49 (+0.28)	0.23 (+0.02)	**0.21**	0.51 (+0.26)	0.26 (+0.01)	**0.25**	0.78 (+0.29)	**0.45** (-0.04)	0.49
alley_2	0.42 (+0.20)	**0.22** (+0.00)	**0.22**	0.48 (+0.23)	0.26 (+0.01)	**0.25**	0.73 (+0.32)	**0.40** (-0.01)	0.41
bamboo_1	0.43 (+0.17)	0.27 (+0.01)	**0.26**	0.45 (+0.17)	**0.28** (+0.00)	**0.28**	0.56 (+0.21)	**0.35** (+0.00)	**0.35**
bamboo_2	0.48 (+0.18)	0.37 (+0.07)	**0.30**	0.45 (+0.11)	0.38 (+0.04)	**0.34**	0.50 (+0.13)	0.41 (+0.04)	**0.37**
bandage_1	1.16 (+0.45)	**0.66** (-0.05)	0.71	1.30 (+0.41)	**0.80** (-0.09)	0.89	1.82 (+0.44)	**1.27** (-0.11)	1.38
bandage_2	0.65 (+0.33)	**0.26** (-0.06)	0.32	0.71 (+0.39)	**0.31** (-0.01)	0.32	0.98 (+0.37)	**0.56** (-0.05)	0.61
shaman_2	0.35 (+0.16)	**0.19** (+0.00)	**0.19**	0.40 (+0.15)	**0.24** (-0.01)	0.25	0.64 (+0.21)	0.45 (+0.02)	**0.43**
shaman_3	0.42 (+0.23)	0.20 (+0.01)	**0.19**	0.51 (+0.29)	**0.22** (+0.00)	**0.22**	0.90 (+0.37)	**0.50** (-0.03)	0.53
temple_2	1.09 (+0.36)	**0.72** (-0.01)	0.73	1.23 (+0.39)	**0.80** (-0.04)	0.84	1.87 (+0.44)	**1.14** (-0.29)	1.43
market_2	1.43 (+0.49)	0.98 (+0.04)	**0.94**	1.40 (+0.35)	**1.01** (-0.04)	1.05	1.58 (-0.08)	**1.24** (-0.42)	1.66
A-EPE(R^{ALL})	0.69 (+0.28)	0.41 (+0.00)	**0.41**	0.74 (+0.27)	**0.46** (-0.01)	0.47	1.04 (+0.27)	**0.68** (-0.09)	0.77
A-EPE(R^{VIS})	0.58 (+0.26)	**0.30** (-0.02)	0.32	0.63 (+0.26)	**0.34** (-0.03)	0.37	0.90 (+0.25)	**0.56** (-0.09)	0.65
A-EPE(R^{FWD})	0.48 (+0.21)	**0.27** (+0.00)	0.27	0.56 (+0.22)	**0.32** (-0.02)	0.34	0.92 (+0.26)	**0.57** (-0.09)	0.66

A. Motion Field Quality

Since the motion field is not known for natural sequences, we use the popular Sintel dataset [14], which consists of a number of highly challenging computer-generated sequences, for which 1-hop *forward* ground truth motion is known. Therefore, while the proposed scheme estimates *both* forward and backward motion fields $M_{k\to k+1}$ and $M_{k\to k-1}$, respectively, which are useful for interpolating extra frames and other tasks, we only measure the quality of the forward motion fields $M_{k\to k+1}$. More precisely, we code the first 21 frames of a number of sequences from the Sintel dataset using HEVC in an IBPB structure, and report results in terms of end-point-error (EPE) of HE-BI on the resulting 10 forward motion fields. The EPE at location $\mathbf{m}$ is computed as

$$\text{EPE}[\mathbf{m}] = \sqrt{(\hat{M}_{\boldsymbol{u}}[\mathbf{x}] - M_{\boldsymbol{u}}[\mathbf{x}])^2 + (\hat{M}_{\boldsymbol{v}}[\mathbf{x}] - M_{\boldsymbol{v}}[\mathbf{x}])^2}, \quad (5)$$

where $\hat{M} = (\hat{M}_{\boldsymbol{u}}, \hat{M}_{\boldsymbol{v}})$ and $M = (M_{\boldsymbol{u}}, M_{\boldsymbol{v}})$ denote the estimated and the ground truth motion fields, respectively. We compute an average EPE in region R^{type}, as

$$\text{A-EPE}(R^{type}) = \frac{\sum_{\mathbf{m}} R^{type}[\mathbf{m}] \cdot \text{EPE}[\mathbf{m}]}{\sum_{\mathbf{m}} R^{type}[\mathbf{m}]}. \quad (6)$$

We define three types of regions; the first one is non-zero only at locations $\mathbf{m}$ where HEVC blocks have a forward motion vector, i.e., $R^{FWD}[\mathbf{m}] = 1$ iff $B_{k\to k+1}[\mathbf{m}] \neq \emptyset$. The second region, R^{VIS}, is non-zero in all regions that are visible in both f_k and f_{k+1}; we take this information from the occlusion masks provided by the Sintel dataset. The last region is the whole domain of the image, i.e., $R^{ALL}[\mathbf{m}] = 1$. We show per-sequence results for R^{FWD}, which is the fairest comparison with HEVC block motion. For R^{VIS} and R^{ALL}, we show the overall A-EPE (i.e., 100 frames).

HE compared to HEVC and original EPIC flow: Table I reports the average *end-point error* (EPE) for three different QP values. Not surprisingly, the A-EPE of HE-BI is

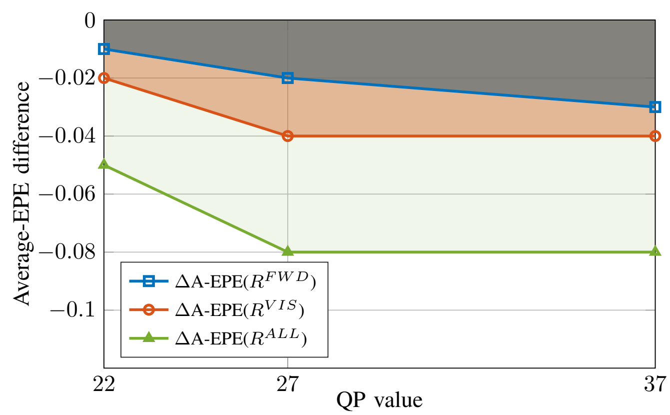
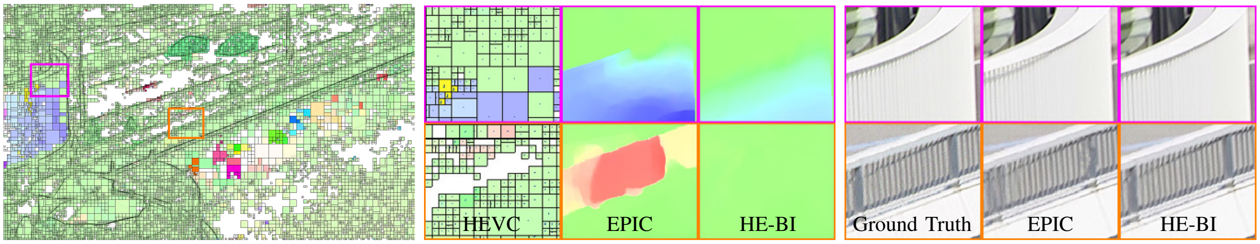


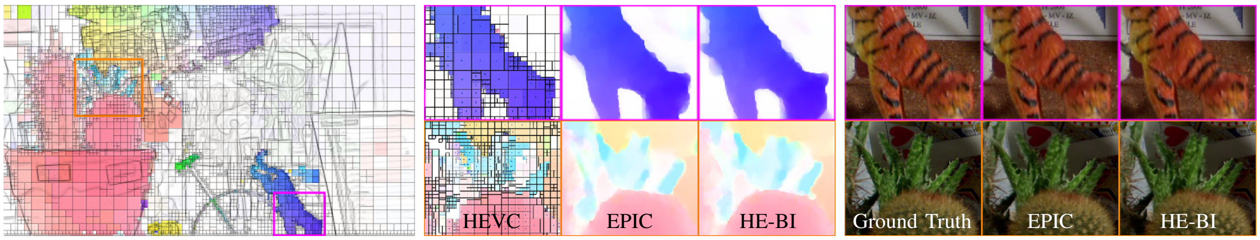
Fig. 4. Overall average-EPE difference (100 motion fields) between HE-BI and HE (ΔA-EPE), for three QP values. More negative numbers mean larger improvements of HE-BI over HE.

significantly lower than for HEVC block motion, especially at high to medium bitrates. Perhaps more surprisingly, the performance of HE-BI is very close to the original EPIC flow algorithm [8]. We reiterate the fact that EPIC employs much more sophisticated correspondences, which are the time-consuming part of this optical flow estimator; on the tested Sintel sequences (1024×416), *HE-BI is over five times as fast* as the original EPIC algorithm; more details on computational complexity can be found in Sect. IV-B. There is a slight drop in performance at lower bitrates, where we observe that the decoded motion blocks are generally quite large. This means that there are fewer seeds available for HE-BI, whereas there will be many more features used by the original EPIC flow algorithm. We plan to address this shortcoming in future work by investigating alternate ways of generating motion seeds from decoded HEVC motion.

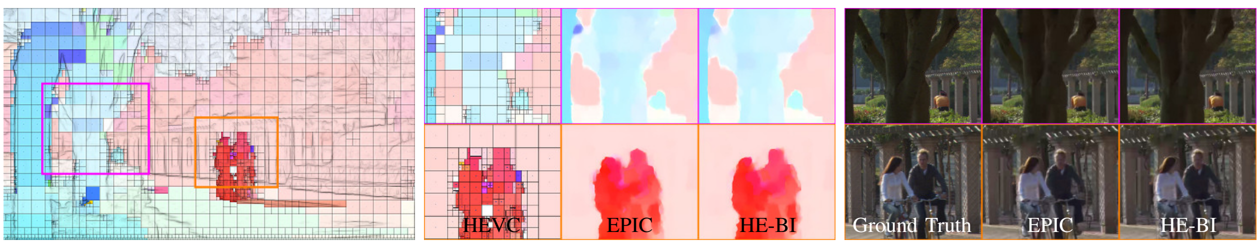
HE versus HE-BI: In Fig. 4, we show the difference of the average EPE between HE and HE-BI (see Sect. III, and



(a) BQTerrace $B_{218\to220}$ (QP=22) (b) BQTerrace $M_{218\to220}$ crops (c) BQTerrace BAM-TFI results for $\hat{f}_{219}$



(d) Cactus $B_{2\to0}$ (QP=27) (e) Cactus $M_{2\to0}$ crops (f) Cactus BAM-TFI results for $\hat{f}_1$



(g) Park $B_{140\to142}$ (QP=37) (h) Park $M_{140\to142}$ crops (i) Park BAM-TFI results for $\hat{f}_{141}$

Fig. 5. TFI performance. (a/d/g) show the decoded block motion, where we overlaid the estimated edge map; (b/e/h) each show crops of the decoded block motion (left), EPIC flow (middle), and proposed HE-BI motion (right); (c/f/i) each show the ground truth target frame, as well as the interpolated frame obtained using EPIC flow (middle), and HE-BI (right) as input motion for BAM-TFI [2].

Fig. 3), averaged over all sequences reported in Table I. In the figure, larger negative values mean bigger improvement of HE-BI over HE.

Our two main observations are:

1) The improvement of HE-BI becomes more apparent at higher QP values, where there are fewer motion vectors available;
2) The biggest improvements can be seen when the EPE is computed over the whole image, i.e., ΔA-EPE(R^{ALL}) (green curve).

The second point is because occluded regions are where the bidirectional completion is most helpful, and these regions are implicitly or explicitly excluded in ΔA-EPE(R^{FWD}) (blue curve) and ΔA-EPE(R^{VIS}) (red curve), respectively.

B. Computational Complexity

In Table II, we report the average timings for the motion estimation of the proposed HE-BI, and compare it to EPIC flow [8]; we report timings at three different resolutions. One can see that EPIC flow spends most of the time finding correspondences using deepmatching (DM) [10]. As mentioned earlier, the appeal of HEVC-EPIC(-BI) is that it avoids the time-consuming finding of correspondences by "recycling" decoded block motion. We note here that we limited the search radius of DM to 64 pixels (so as to match the one used by HEVC), which significantly reduces the computational complexity of DM. Even so, as can be seen in the table, the proposed HE-BI runs over *five times as fast* as EPIC flow.

TABLE II
AVERAGE TIMINGS FOR EPIC FLOW [8], AS WELL AS THE PROPOSED HEVC-EPIC. WE SPLIT THE RESULTS UP IN DEEPMATCHING (DM), EDGE DETECTION (SED), AND EDGE-AWARE INTERPOLATION (INTERP.).

Resolution	DM	SED	Interp.	EPIC flow	HE-BI
1024x416	6.6s	0.25s	1.35s	8.2s	**1.6**s
1280x720	13.5s	0.35s	2.75s	16.6s	**3.1**s
1920x1080	32.1s	0.45s	7.05s	39.6s	**7.5**s

C. Application: Temporal Frame Interpolation (TFI)

In order to show the applicability of the proposed HE-BI motion for video enhancement, we use the motion as input for our recently proposed BAM-TFI scheme [2]. For this experiment, we drop every odd frame, and then encode the resulting subsampled sequence using HEVC in an IBPB structure. The resulting 2-hop block motion is then input

TABLE III
TFI PERFORMANCE COMPARISON (Y-PSNR) OF BAM-TFI [2] WITH PROPOSED HE-BI MOTION AND THE ORIGINAL EPIC FLOW [8] AS INPUT.

	Sequence	Frames	QP=22		QP=27		QP=37	
			EPIC flow	HE-BI	EPIC flow	HE-BI	EPIC flow	HE-BI
720p	Shields	390–410	**34.18** (+0.00)	34.18	33.62 (-0.02)	**33.64**	30.54 (-0.01)	**30.54**
	Stockholm	460–480	33.69 (-0.00)	**33.69**	33.37 (-0.00)	**33.37**	30.38 (-0.01)	**30.40**
	Parkrun	130–150	29.41 (-0.00)	**29.41**	29.01 (-0.00)	**29.01**	26.20 (-0.00)	**26.20**
1080p	BQTerrace	200–220	32.32 (-0.68)	**33.00**	32.59 (-0.46)	**33.05**	**31.59** (+0.03)	31.56
	Cactus	001–021	**32.81** (+0.42)	32.40	**32.56** (+0.35)	32.21	**30.67** (+0.24)	30.43
	ParkScene	136–156	**35.49** (+0.28)	35.20	**34.58** (+0.21)	34.37	**31.35** (+0.17)	31.18
	Station 2	020–040	**38.85** (+0.03)	38.82	**38.25** (+0.10)	38.15	**35.08** (+0.05)	35.03
	Average	-	**33.82** (+0.00)	**33.82**	**33.43** (+0.03)	33.40	**30.83** (+0.07)	30.76

to HE-BI, and the output dense motion is used as input motion field to our recently proposed BAM-TFI scheme [2]. Table III shows the average Y-PSNR for seven standard test sequences (20 interpolated frames each). We observe that at high bit-rates, HE-BI motion performs on par with EPIC motion, whereas at lower bitrates, the Y-PSNR difference slightly increases in favour of EPIC flow; as we have seen in Sect. IV-A, the larger block sizes typically observed at higher QP values result in a lack of seeds for HE-BI. One way of increasing the number of seeds would be to split large blocks up into smaller ones.

The Y-PSNR difference does not have a big impact on the visual quality of the interpolated frames, as evidenced by the qualitative results in Fig. 5. In some cases, as for example for BQTerrace (see Fig. 5c), deepmatching appears to be struggling with the repetitive patterns, which results in erroneous motion and hence visually disturbing artefacts in the interpolated frames. Note how the frame obtained using HE-BI is devoid of these errors.

V. CONCLUSIONS AND FUTURE WORK

This work continues our exploration of obtaining more meaningful motion from decoded HEVC block motion. We consider the case of bidirectional motion attached to B-frames in an IBPB structure. Observing that motion boundaries form a subset of object boundaries, we estimate structured edge information on the decoded target frame. Next, we use the decoded motion vectors as seeds to drive an edge-preserving affine motion interpolation to obtain a dense motion field that is much closer to "physical" motion. In *occluded* regions, we leverage motion information from the other flow direction in order to "complete" the motion field, which results in further improvement of the motion field quality. When used as input to a framerate upsampling scheme, the HE-BI motion produces results that are very close to what state-of-the-art optical flow motion produces, indicating the applicability of HE-BI motion for video enhancement tasks.

In its current implementation, each motion vector is considered as one seed. In future work, we will investigate various seed weighting strategies in order to further improve the results, in particular at lower bitrates.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circ. Syst. for Video Tech.*, vol. 22, no. 12, pp. 1649–1668, 2012.
[2] D. Rüfenacht and D. Taubman, "Temporally Consistent High Frame-Rate Upsampling with Motion Sparsification," *IEEE Int. Workshop on Mult. Sig. Proc.*, 2016.
[3] D. Rüfenacht, R. Mathew, and D. Taubman, "Higher-Order Motion Models for Temporal Frame Interpolation with Applications to Video Coding," *Pict. Cod. Symp.*, 2016.
[4] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Patt. Anal. and Mach. Intell.*, pp. 1744–1757, 2012.
[5] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," *Int. Conf. on Comp. Vis.*, 2015.
[6] Q. Chen and V. Koltun, "Full flow: Optical Flow Estimation by Global Optimization over Regular Grids," *Proc. IEEE Conf. Comp. Vis. and Patt. Rec.*, 2016.
[7] M. Tao, J. Bai, P. Kohli, and S. Paris, "Simpleflow: A non-iterative, sublinear optical flow algorithm," *Computer Graphics Forum*, vol. 31, pp. 345–353, 2012.
[8] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow," *Proc. IEEE Conf. Comp. Vis. and Patt. Rec.*, 2015.
[9] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," *arXiv preprint arXiv:1603.03590*, 2016.
[10] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Deepmatching: Hierarchical deformable dense matching," *Int. J. of Comp. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.
[11] D. Rüfenacht and D. Taubman, "HEVC-EPIC: Edge-Preserving Interpolation of Coded HEVC Motion with Applications to Framerate Upsampling," *Sumbitted to International Conference on Multimedia and Expo*, 2017.
[12] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, 2015.
[13] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B-Pictures and MCTF," *IEEE Int. Conf. on Mult. and Expo*, pp. 1929–1932, 2006.
[14] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," *European Conf. on Comp. Vis.*, 2012.