SIP (2016), vol. 5, e8, page 1 of 14 © The Authors, 2016.

doi:10.1017/ATSIP.2016.8

ORIGINAL PAPER

Occlusion-aware temporal frame interpolation in a highly scalable video coding setting

DOMINIC RÜFENACHT, REJI MATHEW AND DAVID TAUBMAN

We recently proposed a bidirectional hierarchical anchoring (BIHA) of motion fields for highly scalable video coding. The BIHA scheme employs piecewise-smooth motion fields, and uses breakpoints to signal motion discontinuities. In this paper, we show how the fundamental building block of the BIHA scheme can be used to perform bidirectional, occlusion-aware temporal frame interpolation (BOA-TFI). From a "parent" motion field between two reference frames, we use information about motion discontinuities to compose motion fields from both reference frames to the target frame; these then get inverted so that they can be used to predict the target frame. During the motion inversion process, we compute a reliable occlusion mask, which is used to guide the bidirectional motion-compensated prediction of the target 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 which employs piecewise-smooth motion fields with motion discontinuities. We evaluate the proposed BOA-TFI scheme on a large variety of natural and challenging computer-generated sequences, and our results compare favorably to state-of-the-art TFI methods.

Keywords: Temporal frame interpolation, Temporal frame upsampling, Geometrically consistent prediction, Occlusion-handling, Motion discontinuity modeling, Bidirectional hierarchical anchoring of motion fields

Received 29 September 2015; Accepted 7 March 2016

I. INTRODUCTION

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

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. Different frame interpolation (FI) methods have been proposed, which differ in terms of type of motion estimation (ME) performed, as well as where those motion fields are anchored. Also, various strategies and texture optimizations are applied to create the interpolated frame.

A large body of TFI algorithms use block motion fields, which have comparatively low computational complexity. In order to avoid blocking artifacts, various techniques which smooth the interpolated frames are employed. Choi *et al.* [3] use bilateral ME, and block artifacts are reduced using an adaptive overlapped block motion compensation based on the reliability of neighboring motion vectors. Wang *et al.* [4] perform motion-compensated prediction of the intermediate frame from both reference frames independently, and then blend these predictions together using a trilateral filter. Dikbas and Altunbasak [5] use an adaptive interpolation between the forward and backward warped frames. Their method has low computational complexity, but the implicit occlusion handling can lead to severe visual distortions if disoccluded regions become large. Jeong *et al.* [6] perform motion-compensated FI using a multi-hypothesis ME. The best motion hypothesis is selected by optimizing the cost function of a labeling problem. Pixels in the target frame are computed as a weighted combination of several pixels from the reference frame. They show improved reconstruction quality, at

School of EE &T, University of New South Wales, Sydney, Australia

Corresponding author:
D. Rüfenacht
Email: d.ruefenacht@unsw.edu.au

the expense of a significant increase in computational complexity. Veselov and Gilmutdinov [7] propose a hierarchical bidirectional multi-stage ME 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. Zhang *et al.* [8] propose a polynomial motion approximation model in order to account for intensity changes across frames. Their method can be applied to exiting TFI schemes and improve the quality of interpolated frames at the expense of increased memory and computational complexity.

To avoid artificial boundaries created by block motion fields, Chin and Tsai [9] 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. Several FI methods have been proposed which try to detect occluded regions, and show improved performances compared with methods without occlusion handling. Kim *et al.* [10] use linearity checking between the estimated forward and backward motion fields to detect occluded regions. Cho *et al.* [11] use a bidirectional ME scheme that is based on feature trajectory tracking, which allows us to detect occluded regions.

In [12], we have shown how the bidirectional hierarchical anchoring (BIHA) framework naturally lends itself to TFI when all information at a certain temporal level is quantized to zero. The present paper represents an extension of this earlier conference paper. A key distinguishing feature of the proposed *bidirectional, occlusion-aware temporal frame interpolation* (BOA-TFI) framework is that the interpolation process is driven entirely by piecewise-smooth motion estimates that are anchored at reference frames and considered to represent *physical motion*. Motion estimated at reference frames is mapped to target frames where it is used to directly infer regions of occlusion and disocclusion. Motion discontinuities at reference frames are explicitly discovered and play a key role in the motion mapping process. We do not use block motion, as it is ultimately not reflective of the underlying physical reality. We also avoid the use of non-physical averaging processes such as overlapped block motion compensation (OBMC), as employed in most state-of-the-art schemes; such approaches can rarely be justified as modeling an underlying physical process, and often result in oversmooth interpolated frames.

This *motion centric* approach is well adapted to scalable compression schemes [13], because it allows the motion to be understood as part of a transform that is applied to the frame data; the proposed TFI scheme can then be understood as the inverse transform that would result if high temporal frequency details were omitted. We expect that this property will be valuable in enabling seemless integration of video decoding and interpolation processes in the future. In the interest of conciseness, however, this paper focuses only on the TFI problem, leaving the interesting connection with compression to other works.

The motion centric approach means that we do not use texture information (pixel values) as part of the detailed reasoning for the FI process. Frame texture information is used only to derive the piecewise-smooth motion representation itself. Recently, there have been various proposals from the computer vision community on how such motion fields can be estimated. Xu *et al.* [14] propose a motion detail preserving optical flow algorithm (MDP), which encourages sharp discontinuities in the motion field. Wulff and Black [15] propose a layered motion model, which is able to obtain piecewise-smooth motion fields with sharp discontinuities on sequences that are heavily affected by motion blur. While currently limited to two motion layers, this work shows a lot of promise.

With respect to its conference version [12], in this paper we report on improvements that have made the proposed scheme more robust; we also give a much more detailed description of the fundamental concepts of the proposed TFI scheme. Furthermore, we provide a more extensive experimental validation on a large variety of natural sequences, as well as very challenging computer-generated sequences, which turn out to be more difficult than most common natural test sequences. Compared with the prior art, the main differences of the proposed BOA-TFI scheme are:

- High-quality *disocclusion* masks are computed, which are used to guide the *bidirectional prediction* of the interpolated frame – we switch to appropriate unidirectional prediction in regions which are occluded in one reference frame.
- Estimated *motion field discontinuity* information allows us to reliably identify the foreground object in regions of motion field folding (i.e., resolve double mappings).
- If used with our *highly scalable video coding* (HSVC) scheme, motion fields which were estimated at the *encoding stage* can be (re)used for FI, which significantly reduces the computational complexity at the decoder. Additionally, motion fields can be estimated on high-quality texture data at the encoder, as opposed to decoded frames which might suffer from compression artifacts.

II. OVERVIEW

Figure 1 gives an overview of the proposed TFI method, where we used the notation introduced in Table 1. Inputs to the method are the reference frames f_a and f_c, and the (potentially estimated[1]) motion field between them, $M_{a\to c}$. The proposed scheme involves two types of operations on motion fields: motion *inference* and motion field *inversion*. Both these operations involve mapping a motion field from one frame to another, which likely leads to double mappings (because of folding of the motion field), as well as holes in regions that get disoccluded. Both double mappings and disoccluded regions are handled by reasoning about the displacement of motion discontinuities.

A variety of ways can be employed to represent motion discontinuities in the proposed framework. Because this

[1]We describe a way of estimating piecewise-smooth motion fields suitable for this work in Section VII.

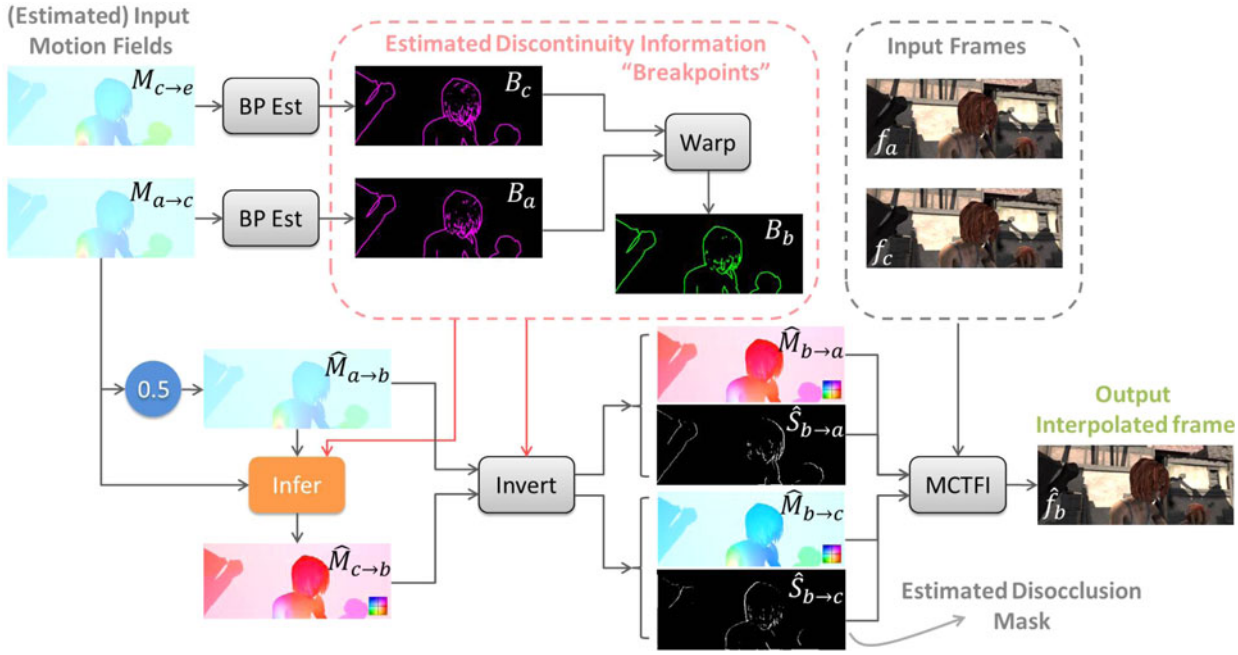


Fig. 1. Overview of the proposed TFI method: The input to the scheme are a (potentially estimated) motion field $M_{a\to c}$, as well as breakpoint fields *estimated* on $M_{a\to c}$ for frame f_a, and on $M_{c\to e}$ (only used to obtain breakpoints) for frame f_c; furthermore, the two reference frames f_a and f_c. In the first step, estimated breakpoints at reference frames f_a and f_c (B_a and B_c) are transferred to the target frame f_b (B_b). Next, $M_{a\to b}$ is obtained by halving its parent motion field $M_{a\to c}$. $M_{a\to c}$ and $M_{a\to b}$ are then used to *infer* the motion field $M_{c\to b}$. The last step consists of *inverting* $M_{a\to b}$ and $M_{c\to b}$ to obtain $\hat{M}_{b\to a}$ and $\hat{M}_{b\to c}$. During the motion inversion process, we compute disocclusion masks $\hat{S}_{b\to a}$ and $\hat{S}_{b\to c}$, which are used to guide the bidirectional MCTFI process to temporally interpolate the frame $\hat{f}_b$. Breakpoints are used to resolve double mappings and handle occluded regions during *both* the motion *inference* and *inversion* process.

Table 1. Table of notations used throughout the paper.

Notation	Meaning
f_a, f_c	Reference frames
f_b	Target frame (frame to be interpolated)
$M_{i\to j}$	Motion field anchored at frame f_i and pointing to frame f_j
B_i	Motion discontinuity information (represented using breakpoints) anchored at frame f_i
$S_{i\to j}$	Disocclusion mask anchored at frame f_i; its values are non-zero only at locations that are *not* visible in frame f_j
$\hat{A}$	Used to denote an *estimate* of an entity A, e.g. $\hat{M}_{i\to j}$ denotes an estimated motion field

work comes out of a highly scalable video coder [13], we use *breakpoints* to represent *motion discontinuities* [16]; breakpoints are very useful in a scalable video coder because of their high scalability attributes both in quality and resolution. See Section IV for more details on how breakpoints are employed in this work to induce motion discontinuities.

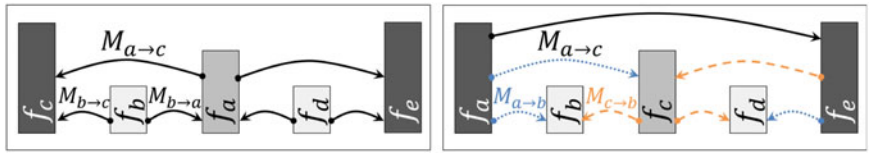
The first step of the proposed method consists of warping motion discontinuity information from reference frames f_a and f_c, to the (non-existent) target frame f_b. Next, we compute an estimate $\hat{M}_{a\to b}$ of the motion field between frame f_a and the target frame f_b by *scaling* the parent motion field $M_{a\to c}$ by a factor of 0.5. Next, we *infer* a motion field $\hat{M}_{c\to b}$, which is anchored at frame f_c and pointing backwards to frame f_b; to infer $\hat{M}_{c\to b}$, both its temporal parent motion $M_{a\to c}$, and its temporal sibling $\hat{M}_{a\to b}$, are used.

The next step is to *invert* both $\hat{M}_{a\to b}$ and $\hat{M}_{c\to b}$, so that we obtain the two motion fields $\hat{M}_{b\to a}$ and $\hat{M}_{b\to c}$, which are anchored at the target frame f_b we want to interpolate. During this inversion process, we readily observe regions of the motion that are getting disoccluded; such regions are recorded in the disocclusion masks $\hat{S}_{b\to a}$ and $\hat{S}_{b\to c}$, and are used to guide the *bidirectional, occlusion-aware* motion-compensated temporal frame interpolation (MCTFI) process.

III. BIDIRECTIONAL HIERARCHICAL ANCHORING OF MOTION FIELDS

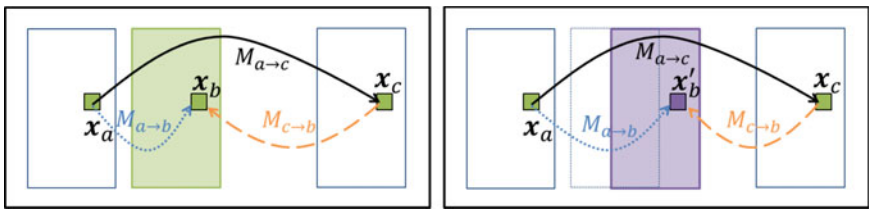
All current state-of-the-art video codecs anchor motion fields at the *target* frames. In [17], we proposed to anchor motion fields at the reference frames instead. In this paper, we demonstrate how the underlying methods of constructing motion fields are highly suited for FI, and can lead to a geometrically consistent bidirectional prediction of the interpolated target frames. Perhaps surprisingly, these interpolated frames can have higher quality than those produced by state-of-the-art TFI schemes. Figure 2 shows the two different ways of anchoring motion fields.

Let us assume that all *odd* frames (f_b and f_d in Fig. 2) 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



(a) Traditional Anchoring (b) BIHA

Fig. 2. (a) Traditional anchoring of motion fields in the target frames and (b) bidirectional hierarchical anchoring (BIHA) of motion fields at reference frames.



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

Fig. 3. 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 $\hat{M}_{a\to b}$ and $\hat{M}_{c\to b}$ are geometrically consistent.

decoder. In our scalable video coding scheme, we closely model "true" motion fields, which can be *scaled* and hence readily be used to perform FI at the decoder.

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 Section V.A. Around the 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, that 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 avoid this strategy for two main reasons:

(i) In a highly scalable video coder, this would be redundant information.
(ii) It is very likely that $M_{a\to c} \neq (M_{c\to a})^{-1}$, in particular around the moving objects. Hence, their scaled versions will not be geometrically consistent in frame f_b.

We instead *infer* $\hat{M}_{c\to b}$, anchored at frame f_c, from the forward pointing motion field $M_{a\to c}$ and its *scaled* version $\hat{M}_{a\to b}$, as follows:

$$\hat{M}_{c\to b} = \hat{M}_{a\to b} \circ (M_{a\to c})^{-1}, \tag{1}$$

where $\circ$ denotes the *composition* operator. 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. 3. In practice, what this means is that the predicted target frame will be significantly less blurred and contain less ghosting than traditional TFI approaches; for examples, the reader is referred to Fig. 11. We remind the reader that a key principle in this work is to *avoid averaging techniques* (such as OBMC) that do not correspond to physical motion.

IV. 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 (see Section V.A), as well as to extrapolate motion in disoccluded regions during the motion field *inference* process to obtain $\hat{M}_{c\to b}$ (see Section V.B). As this work builds upon an HSVC framework, we use a highly scalable way of coding discontinuities using *breakpoints*[2], where they are used to modify the behavior of the discrete wavelet transform (DWT) in the vicinity of (motion) discontinuities. In essence, the presence and precision of breakpoints in the hierarchical representation is determined in a rate-distortion optimized way; the interested reader is referred [16] for a much more detailed description of the technical details on the estimation of breakpoints. In the following, we give a brief summary of how breakpoints are used to induce motion discontinuities. We then present how breakpoints can be transferred from reference frames to the target frame we want to interpolate.

[2] In a scalable video coding framework, the piecewise-smooth motion fields are constructed by the discontinuities and smooth data. For conciseness, we leave the interesting connection between scalable video compression and TFI for future work.

A) Inducing motion discontinuities from breakpoints

This section presents how motion discontinuity information can be induced from an existing breakpoint field; for a comprehensive description of how breakpoints used in this work are estimated, we refer the interested reader to [16]. Breakpoints lie on grid *arcs*, and can be connected to form discontinuity line segments. They are organized in a hierarchical manner, such that breakpoints at finer spatial levels can be *induced* from coarser levels. We use Fig. 4 to guide the description.

A breakpoint field at spatial level η consists of *cells* of size $2^\eta \times 2^\eta$ pixels; these cells are the fundamental unit used to induce discontinuities. A cell consists of four *perimeter arcs* (cyan lines in Fig. 4), as well as two *root arcs* (gray lines in Fig. 4). The significance of root arcs is that they do not exist at coarser levels in the pyramid. Each arc can be occupied by at most one breakpoint. If a cell contains exactly two perimeter breakpoints, and the root arcs at this level have no explicitly coded breaks, connecting the two perimeter breaks allows breakpoints to be induced onto the root arcs. To avoid confusion, we use the term *vertices* to identify the explicitly coded breaks. What this means then is that spatial induction transfers discontinuity information recursively from coarser level vertices to finer levels in the hierarchy, except where such transfer would be in conflict with finer level vertices.

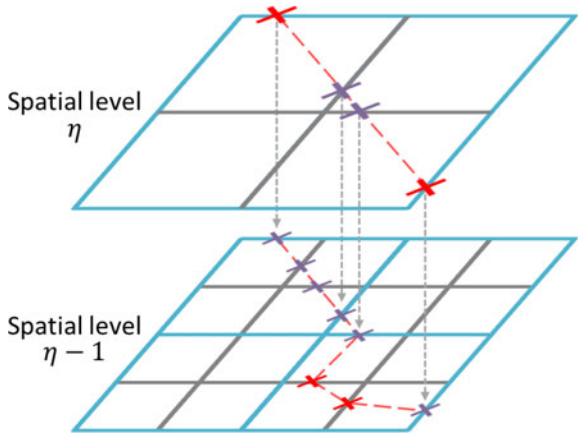


Fig. 4. Scalable geometry representation: Two breakpoints on the *perimeter* of the same *cell* can induce discontinuity information onto the *root* arcs (purple crosses). If the root arc contains a vertex (red cross), the inducing is stopped.

B) Temporal breakpoint induction

For TFI, 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 [18]. The underlying idea of mapping breakpoints from reference to target frames is the fact that motion discontinuities travel with the foreground object. Because the presence of a breakpoint necessarily implies that the motion on either side of it is significantly different, the aim is to identify the foreground motion by performing a breakpoint compatibility check between the two reference frames f_a and f_c, and then to warp compatible line segments to the target frame by halving the identified foreground motion. Figure 5 illustrates the three main steps of the proposed *hierarchical* temporal breakpoint induction method:

(i) Breakpoint compatibility check to find *compatible* (i.e., foreground) motion to assign to discontinuity line segments.
(ii) 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*).
(iii) Upsampling of breakpoints to the next finer spatial resolution (*spatial induction*).

In cases where a warped line segment intersects an arc that already contains a spatially induced breakpoint, the temporally induced breakpoint always overwrites the spatially induced one.

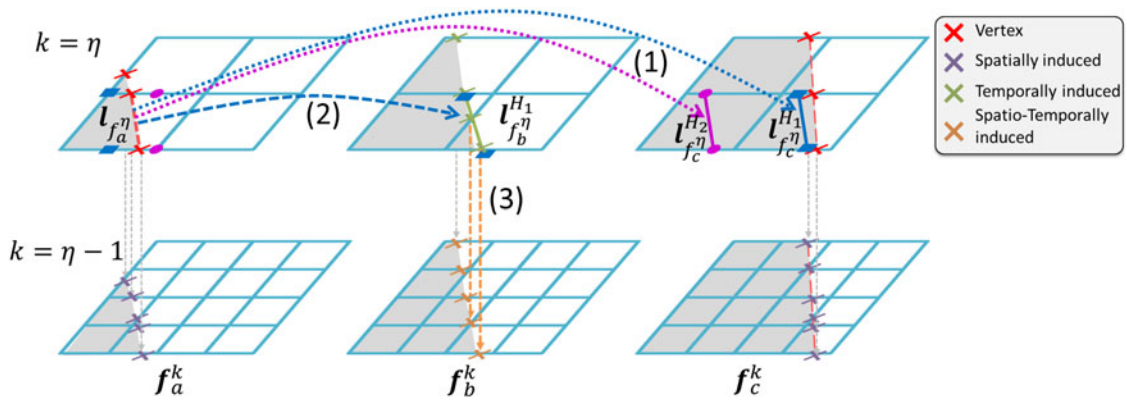
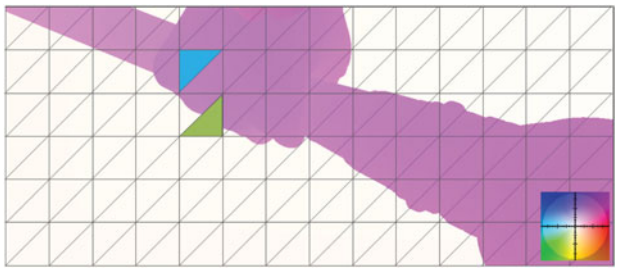
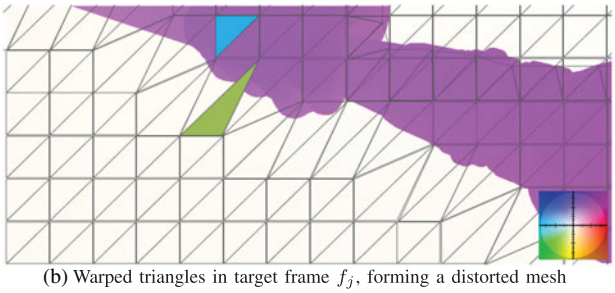


Fig. 5. Spatio-temporal induction of breakpoints. Going from coarse to fine spatial resolution, the proposed temporal induction process consists of three steps at each resolution level η: (1) Assessment of temporal compatibility of line segments induced by breakpoints between two coarse-level frames f_a and f_c; (2) Warping of compatible line segments to f_b; (3) Spatial induction of all breakpoints to the next finer spatial resolution $\eta - 1$. For better visualization, root arcs are not shown in this figure.

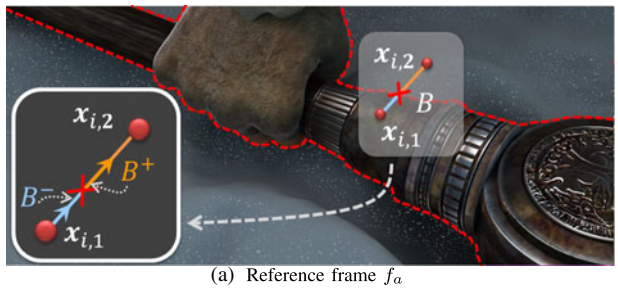


(a) Triangular partition of the reference motion field $M_{i\to j}$

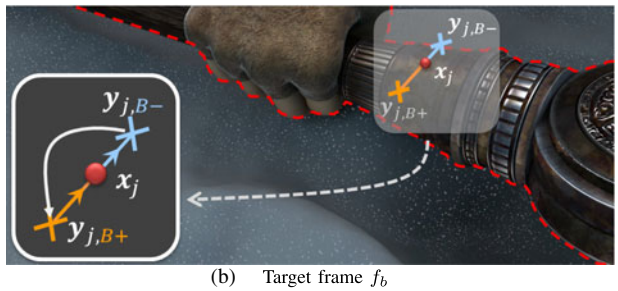


(b) Warped triangles in target frame f_j, forming a distorted mesh

Fig. 6. Illustration of the proposed CAW procedure. These figures show color–coded motion fields. (a) The reference motion field is partitioned into triangles; (b) each such triangle is then mapped from the reference to the target frame, where each integer location gets assigned the corresponding affine motion. In regions that get disoccluded, triangles stretch without changing orientation (e.g., the green triangle), and the affine model assigns a interpolated value between the foreground and background motion, without leaving any hole.



(a) Reference frame f_a



(b) Target frame f_b

Fig. 7. Resolving of double mappings in the mapped motion field by reasoning about motion discontinuities (represented as red dashed lines around the scepter). The key idea in identifying the foreground motion is that the motion discontinuities travel with the foreground object.

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.

V. MOTION FIELD OPERATIONS

In this section, we present *two* motion field operations that are used in the proposed TFI method, and show how motion discontinuity information is used to solve key problems current TFI methods suffer from, namely the handling of double mappings, as well as occluded regions.

A) Inversion of motion fields

Most TFI methods map either pixels or whole blocks from reference to target frames, which creates a variety of unwanted artifacts such as holes within objects because the adjacent blocks have different motion assigned. Also, even if ground truth motion were used, a simple mapping can lead to holes in the target frame if the object is expanding.

To avoid these problems, we employ piecewise-smooth motion, and employ a *cellular affine warping* (CAW) procedure first proposed in [18] to warp motion fields from one frame to another. We use Fig. 6 to guide the description of the CAW procedure. In the current implementation, the reference motion field is partitioned into triangles of size 1×1 pixel, so that there are approximately twice as many triangles as there are pixels in the frame.[3] The warped motion field is guaranteed to have no holes (in disoccluded regions). On the leading side of moving objects, one is likely to observe double mappings during the motion field warping process. In the following, we explain how such double mappings can be resolved using motion discontinuity information.

1) Identifying foreground motion in double-mapped regions

As explained in the previous section, as the CAW procedure maps triangles from reference to target frames, in regions of folding, multiple triangles map to the same location $\mathbf{x}_j$ in the target frame f_j. In other words, there are two locations $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ in f_i, which are mapped by $M_{i\to j}$ to the same location. In this section, we show how *motion discontinuity* information can be used to locally reason about foreground moving objects. We use Fig. 7 to guide the description. We denote the line segment that connects $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ in the reference frame f_i as $\boldsymbol{l}$; this line has to intersect with (at least) one motion discontinuity, denoted as B in the figure. In the example, the scepter is lifted and moves on top of the snow in the background. Let B^- denote the location on $\boldsymbol{l}$ which is on the same side as $\mathbf{x}_{i,1}$; similarly, let B^+ denote the location on $\boldsymbol{l}$ that is on the side of $\mathbf{x}_{i,2}$. Because the motion discontinuity moves with the foreground object, either $\mathbf{y}_{j,\boldsymbol{B}^-} = M_{i\to j}(\boldsymbol{B}^-)$ or $\mathbf{y}_{j,\boldsymbol{B}^+} = M_{i\to j}(\boldsymbol{B}^+)$ will map very closely to a motion discontinuity in the target frame f_j; this is the foreground motion we register. In the example in the figure, $\mathbf{y}_{j,\boldsymbol{B}^-}$ gets mapped onto motion discontinuities; therefore, $\hat{M}_{a\to b}(\boldsymbol{x}_{i,1})$ gets recorded as foreground motion

[3] Clearly, this procedure can be made much more efficient by increasing the triangle size in regions of smooth motion.

at location $\boldsymbol{x}_j$ where the double mapping occurred (e.g., $\hat{M}_{b\to a}(\boldsymbol{x}_j) = -\hat{M}_{a\to b}(\boldsymbol{x}_{i,1})$).

2) Obtaining a disocclusion mask

The inversion of $M_{i\to j}$ allows us to readily observe regions that get disoccluded in the target frame; we record this valuable information in a *disocclusion mask* $S_{j\to i}$ as follows:

$$S_{j\to i}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \text{ disoccluded,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In the proposed bidirectional prediction setup, we obtain two such *disocclusion masks* anchored at the target frame f_b: one during the inversion of $M_{a\to b}$, which we denote $S_{b\to a}$, and the other $S_{b\to c}$, obtained during the inversion of $M_{c\to b}$. They are used to generate the interpolated frame as explained in Section VI.

B) Motion field inference

As shown in equation (1), the backward pointing motion field $\hat{M}_{c\to b}$, anchored at frame f_c, is *inferred* from the forward pointing motion field $M_{a\to c}$ and its *scaled* version $\hat{M}_{a\to b}$. As mentioned earlier, one advantage of this operation is that $\hat{M}_{a\to b}$ and $\hat{M}_{c\to b}$ are *geometrically consistent*, meaning that the interpolated target frame will contain much less ghosting artifacts.

Both $\hat{M}_{a\to b}$ 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). Part of the motion field inference process involves the inversion of the motion field $M_{a\to c}$; during this process, we readily observe regions that are not visible in f_a. The CAW procedure assigns a linear interpolation between background and foreground motion to disoccluded regions; in order to be most useful, however, motion in disoccluded regions of $\hat{M}_{c\to b}$ should be extrapolated from the triangle vertices falling on one side of the motion discontinuities. In the following, we describe this procedure in more detail.

1) Motion extrapolation in disoccluded triangles

The aim of the motion inference process is to obtain a motion field $\hat{M}_{c\to b}$, anchored at frame f_c, and pointing to f_b, which is as close to a "real" motion field as possible. In the absence of new motion appearing in regions that get disoccluded between frames f_a and f_c, a good estimate for the motion is to extrapolate the motion of the triangle vertices up to motion discontinuity boundaries. For most of the disoccluded triangle, this means that background motion is extrapolated; only a small (if any) part of the triangle falls onto the foreground object. We use Fig. 8 to explain the details of the proposed motion extrapolation technique.

Whenever a triangle is stretching as it is mapped from a reference to a target frame, we expect it to intersect with motion discontinuities in the target frame; this is because some of its vertices belong to the background (possibly in motion), and some belong to the foreground (moving) object. In Fig. 8(c), D'_1 and D'_2 sit in the background, whereas D'_3 belongs to the foreground. The warped triangle has two edges that intersect with motion discontinuities, which we denote as e_1 and e_2. As mentioned before, instead of interpolating a value transitioning from background (D'_1 in Fig. 8) to the foreground motion D'_3, we want to extrapolate the background motion up to the motion boundary, and likewise extrapolate the foreground motion up to the motion boundary. To clarify this, we show a 1D cut along the e_1, formed by connecting D'_1 and D'_3, of the horizontal component (mv_x) of the motion in Fig. 8(d); the dashed blue line shows the motion assigned by the CAW procedure, and the green solid (staircase) shows the background and foreground extrapolated motion. Irrespective of what object (foreground or background) each of the three vertices of the triangle belongs to, the motion extrapolation method performs the same steps: The motion of D'_3 is extrapolated in the triangle formed by D'_1, B_1, and B_2. The quadrilateral (D'_1, D'_2, B_1, B_2) is broken up into two triangles (D'_1, D'_2, B_1) and (D'_2, B_1, B_2), and the motion of D'_1 and D'_2 is extrapolated in the respective triangles.

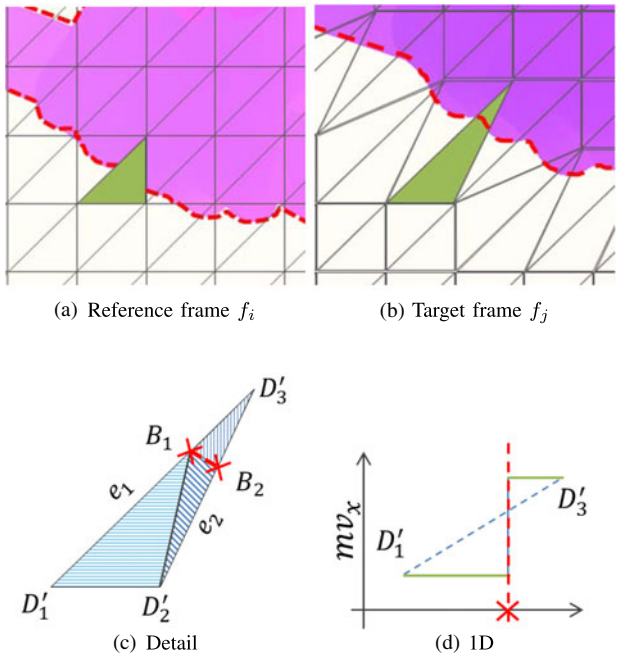
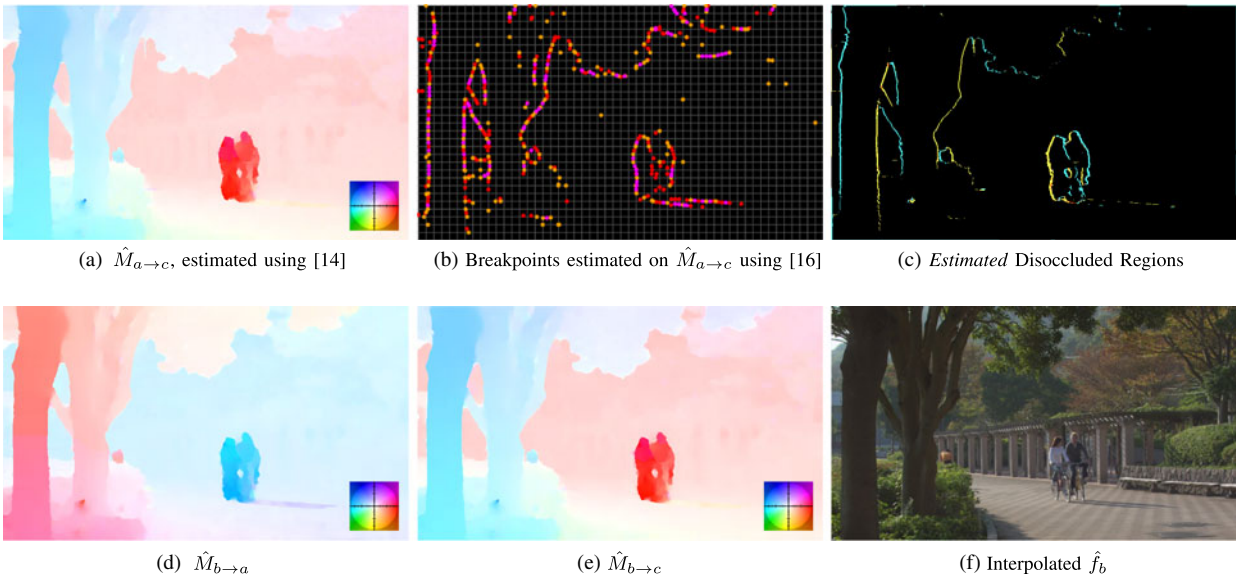


Fig. 8. Closeup of the scene in Fig. 6, to illustrate the motion extrapolation technique applied in disoccluded regions. Panel (a) shows a triangle in the reference frame f_i, which straddles a motion discontinuity boundary. Panel (b) shows the warped, stretched triangle in the target frame f_j; panel (c) introduces the relevant notations used in the text to describe the motion extrapolation procedure. Instead of linearly interpolating motion from foreground to background, we instead extrapolate motion from the vertices to the motion discontinuity boundary, represented by B_1 and B_2; this results in sharp boundaries, as exemplified in (d), where the blue dotted line corresponds to linearly interpolated motion, and the green solid line corresponds to extrapolated motion.

VI. Motion-Compensated Temporal 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



(a) $\hat{M}_{a\to c}$, estimated using [14] (b) Breakpoints estimated on $\hat{M}_{a\to c}$ using [16] (c) *Estimated* Disoccluded Regions

(d) $\hat{M}_{b\to a}$ (e) $\hat{M}_{b\to c}$ (f) Interpolated $\hat{f}_b$

Fig. 9. Example results for estimated motion on a natural sequence with reasonably complex motion. Panel (a) shows the motion field $\hat{M}_{a\to c}$, estimated using [14] with the default parameters; Panel (b) shows the breakpoint field (at the second coarsest spatial level for visualization), which was estimated on $\hat{M}_{a\to c}$ using the breakpoint estimation method described in [16]. Panel (c) shows the union of the estimated disocclusion masks, where yellow and cyan indicate that the pixel is not visible in the previous (f_a) and future (f_c) frame, respectively. Panels (d) and (e) show the inverted motion fields, anchored at the target frame f_b, which together with the disocclusion mask are used to obtain (f), the bidirectionally predicted target frame $\hat{f}_b$.

frame f_i. The warping of frame f_j to frame f_i, evaluated at location $\mathbf{x}$, is then denoted as $f_{j\to i}(\mathbf{x}) = (\mathcal{W}_{\hat{M}_{i\to j}}(f_j))(\mathbf{x})$. 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:

$$\hat{f}_b(\mathbf{x}) = \begin{cases} \dfrac{S_{b\to a}(\mathbf{x})\, f_{a\to b}(\mathbf{x}) + S_{b\to c}(\mathbf{x})\, f_{c\to b}(\mathbf{x})}{\kappa(\mathbf{x})} & \kappa(\mathbf{x}) > 0, \\ 0.5\big(f_{a\to b}(\mathbf{x}) + f_{c\to b}(\mathbf{x})\big) & \kappa(\mathbf{x}) = 0, \end{cases} \tag{3}$$

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

Regions in f_b which are disoccluded in both of the reference frames (i.e., $\kappa(\mathbf{x}) = 0$), are predicted from both reference frames equally, where the affine warping process results in a stretching of the background texture information.

VII. ESTIMATION OF PIECEWISE-SMOOTH MOTION FIELDS WITH DISCONTINUITIES

In the proposed work, we require piecewise-smooth motion fields with sharp discontinuities at moving object boundaries. The estimation of such motion fields that are tailored for the proposed scheme is a parallel, ongoing stream of research. To show the applicability of the proposed scheme on natural sequences, we need to estimate motion fields that satisfy our requirements. We found that Xu *et al.*'s [14] motion detail preserving (MDP) optical flow algorithm provides motion fields of sufficient quality to work with our proposed framework; the parent motion field $\hat{M}_{a\to c}$ is estimated using the default parameters of their implementation.

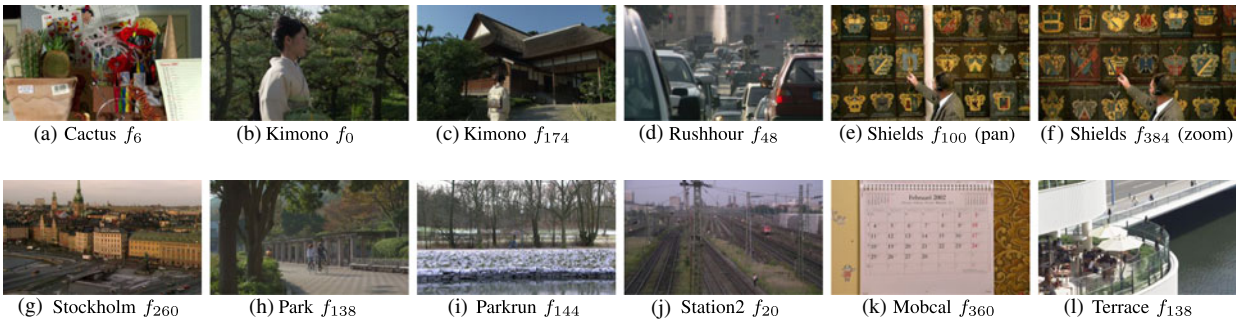
MDP uses an extended coarse-to-fine refinement framework, which is able to recover motion details at each scale by reducing the reliance of flow estimates that are propagated from coarser scales. Large displacements are handled by using sparse feature detection and matching, and a dense nearest-neighbor patch matching algorithm is used to handle small textureless regions which are likely missed by the feature matching algorithm. Furthermore, an adaptive structure map which maintains motion discontinuity is used in the optical flow regularization term.

Next, we run Mathew *et al.*'s [16] breakpoint estimation scheme to estimate motion discontinuities on $\hat{M}_{a\to c}$ (see Section IV). Figure 9 shows an example estimated motion and breakpoint field. To show the applicability of the estimated motion and breakpoint field on natural sequences, we further show estimated disocclusion masks, *inverted* motion fields, as well as temporally interpolated frame $\hat{f}_b$.

We note that in our wavelet-based highly scalable video coder [13], the motion field estimation and the breakpoint estimation is performed at the encoder; at the decoder, only the motion field inversion and subsequent motion-compensated prediction of the frame to be interpolated have to be performed, which significantly reduces the computational complexity of the proposed approach.

VIII. EXPERIMENTAL EVALUATION AND DISCUSSION

In our previous work [12], we have shown preliminary results of the proposed BOA-TFI method on synthetic sequences, and have highlighted the quality of the proposed method in occluded regions. In this work, we significantly



(a) Cactus f_6 (b) Kimono f_0 (c) Kimono f_{174} (d) Rushhour f_{48} (e) Shields f_{100} (pan) (f) Shields f_{384} (zoom)

(g) Stockholm f_{260} (h) Park f_{138} (i) Parkrun f_{144} (j) Station2 f_{20} (k) Mobcal f_{360} (l) Terrace f_{138}

Fig. 10. First frame of each of the natural sequences used in the experiments. All sequences are readily available on https://media.xiph.org/video/derf/.

enhance the evaluation of the proposed method both qualitatively and quantitatively on various *high-resolution* natural sequences (Section VIII.A), and compare our performance with two state-of-the-art TFI methods [6, 7].

One key distinguishing feature of the proposed method is its ability to handle regions around moving objects. Video resolution has seen a significant increase in recent years, while the frame-rate has not dramatically changed; what this means is that the expected size of disoccluded regions is larger, which makes appropriate handling of such regions more important. By contrast, the handling of occluded regions on low-resolution video (e.g., CIF and lower) is not so important, since they tend to be small. On such low-resolution sequences, our TFI method performs similarly to existing TFI methods, and sometimes even worse, because we do not apply any smoothing to our interpolated frames. In this paper, we want to highlight the importance of better motion and interpolation methods for high-resolution data; for this reason, all experiments are performed on high-resolution video sequences.

Recently, the computer-generated animation movie "Sintel" has become very popular in the computer vision community because of its complexity and high correlation with natural sequences [19]; in Section VIII.C, we further show qualitative results of the proposed motion inference scheme on various scenes from the Sintel sequence, which contain much larger amounts of disocclusions than the natural sequences.

A) Results on natural sequences

In this section, we show the results obtained on common test sequences; motion fields are estimated using the optical flow estimator proposed by Xu *et al.* [14], as detailed in Section VII. We compare our results with two state-of-the-art TFI methods: Jeong *et al.* [6] focus on a sophisticated multi-hypothesis testing framework, where a lot of effort is spent on texture optimization. Veselov and Gilmutdinov [7] focus on estimating high-quality motion fields, which are then used without any sophisticated texture optimization to interpolate the target frame.

We selected 12 sets of various common high-resolution test sequences with a large variety of motion and texture complexity; Fig. 10 shows the first frame of each sequence. For each such sequence, we choose 11 adjacent *even* numbered frames, and interpolate the *odd* numbered frames in between them; this results in 10 interpolated frames per sequence. Table 2 presents the per sequence results, averaged over the 10 frames.

Table 2. Quantitative comparison of the proposed method with [6, 7] on common natural test sequences. In parantheses $(\cdot)$, we show the difference between the PSNR of the proposed BOA-TFI method and the respective method we compare it with ("−" means that the proposed BOA-TFI performs better, "+" means worse performance).

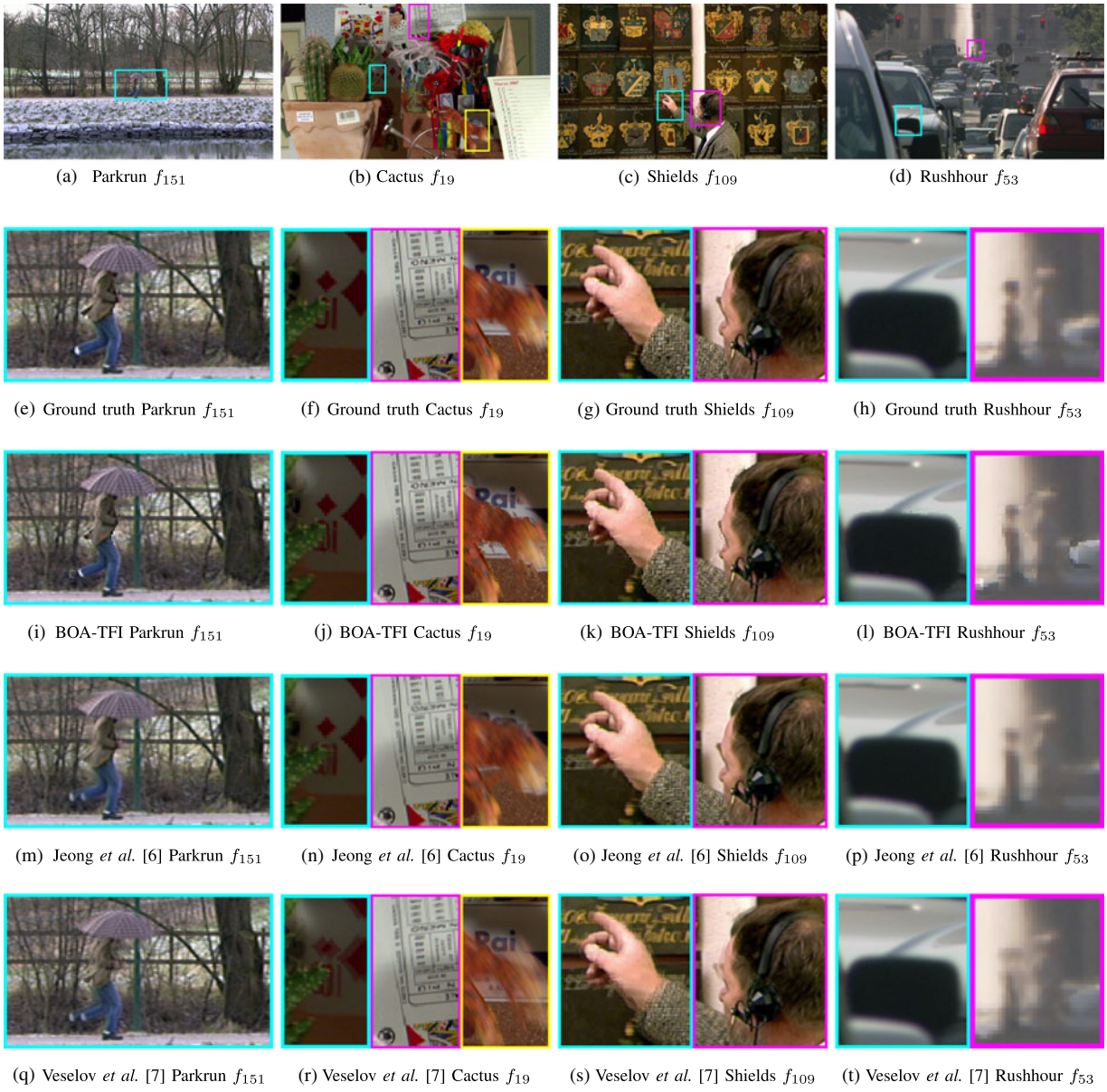
Sequence	Frames	Jeong *et al.* [6]	Veselov and Gilmutdinov [7]	BOA-TFI
Cactus	007–025	33.15 (−0.52)	31.28 (−2.39)	**33.68**
Kimono	001–019	**33.92** (+0.68)	33.39 (+0.14)	33.25
Kimono	175–193	39.94 (−0.83)	40.14 (−0.63)	**40.77**
Rushhour	049–067	**35.16** (+0.52)	34.91 (+0.27)	34.64
Shields	101–119	35.90 (−0.16)	35.06 (−0.99)	**36.05**
Shields	385–403	33.87 (−3.91)	35.55 (−2.22)	**37.77**
Stockholm	261–279	36.58 (−1.26)	37.09 (−0.75)	**37.84**
Park	139–157	38.26 (−0.98)	38.81 (−0.43)	**39.24**
Parkrun	145–163	30.62 (−1.20)	30.95 (−0.87)	**31.82**
Station2	021–039	40.05 (−1.79)	40.91 (−0.93)	**41.84**
Mobcal	361–379	29.13 (−8.59)	34.75 (−2.98)	**37.73**
Terrace	139–157	33.27 (−4.36)	34.20 (−3.43)	**37.63**
Average	–	34.99 (−1.87)	35.59 (−1.27)	**36.85**

Bold indicates the best performance for a given sequence.

While reporting average peak signal-to-noise ratio (PSNR) values provides a compact way of summarizing the performance of the tested methods, we note that this measure only makes sense in regions where there is no acceleration between the two reference frames. Ultimately, it is the perceived visual quality that is important. We therefore provide qualitative results for some of the sequences in Fig. 11. First off, both TFI methods chosen for comparison are able to provide high-quality interpolated frames, in particular in regions inside moving objects (i.e., away from moving object boundaries). The differences in PSNR values and visual quality are governed by *two major factors*:

1) How regions of global motion are interpolated

Block-based methods usually employ a variant of OBMC, which tends to oversmooth the interpolated frames, resulting in significant blurring of the overall texture. In Fig. 11,



(a) Parkrun f_{151} (b) Cactus f_{19} (c) Shields f_{109} (d) Rushhour f_{53}

(e) Ground truth Parkrun f_{151} (f) Ground truth Cactus f_{19} (g) Ground truth Shields f_{109} (h) Ground truth Rushhour f_{53}

(i) BOA-TFI Parkrun f_{151} (j) BOA-TFI Cactus f_{19} (k) BOA-TFI Shields f_{109} (l) BOA-TFI Rushhour f_{53}

(m) Jeong *et al.* [6] Parkrun f_{151} (n) Jeong *et al.* [6] Cactus f_{19} (o) Jeong *et al.* [6] Shields f_{109} (p) Jeong *et al.* [6] Rushhour f_{53}

(q) Veselov *et al.* [7] Parkrun f_{151} (r) Veselov *et al.* [7] Cactus f_{19} (s) Veselov *et al.* [7] Shields f_{109} (t) Veselov *et al.* [7] Rushhour f_{53}

Fig. 11. Qualitative comparison of TFI on natural sequences. The first row shows the full frame. The second to last row show crops of the ground truth, proposed BOA-TFI, Jeong *et al.* [6] and Veselov and Gilmutdinov [7], respectively.

this can be seen in highly textured regions such as the running man with the umbrella in the first column, as well as the text on the card of the Cactus sequence in the second row.

2) How regions around moving objects are handled

Regions around moving objects are only visible from one reference frame, and hence should only be predicted from the frame they are visible. This can only be achieved if such regions are detected. The quality of the proposed occlusion handling can be appreciated in various crops shown, but is most visible in the "Parkrun" sequence, as well as the "Cactus" sequence, where the "10" (cyan crop) is properly interpolated by our method.

In the current implementation of the proposed method, we do not perform any texture optimization. In regions which are highly affected by motion blur, such as the tiger in the "Cactus" sequence, this can create artificial high frequencies. A similar observation is also noted for the "Rushhour" sequence, which is highly affected by motion blur and atmospheric blur. For the first frames of the "Kimono" sequence, the optical flow estimator has problems on the right side of the woman, and mistakenly associates background pixels to the foreground object. While hardly visible, this results in a significant PSNR drop.

We plan to address the above-mentioned problems in future work by selectively smoothing the prediction in regions where there is a transition from uni- to bidirectional prediction; such regions can easily be identified by the presence of motion discontinuities.

Table 3. Average per-frame processing time (in sec) on all the frames tested in Section VIII.A, split up in ME and FI, as well as total time. We further provide the CPU and amount of RAM of the machines the results were obtained.

Method	Language	CPU (GHz)	RAM (GB)	ME	FI	Total
Jeong *et al.* [6]	C	2.8	8	410.2	498.9	909.1
Veselov and Gilmutdinov [7]	Matlab	2.6	8	32.4	2.1	34.5
BOA-TFI	C	3.2	8	355.4	8.2	363.6

B) Processing times

In this section, we report on the processing times of the proposed TFI method, and compare it with [6, 7]. It is important to note that none of the methods is optimized for time, and the timings were obtained on different machines. Table 3 shows the relevant specifications of the testing machines, as well as the average per-frame processing time. As mentioned before, we use [14] to estimate motion fields, and the contribution of this work is how such motion and estimated motion discontinuity information can be used to improve the FI process. For this reason, we split up the processing times for the ME part and the FI part.

One can see that most of the processing time in the proposed BOA-TFI method is spent on estimating the motion, which is currently done using [14]. Veselov and Gilmutdinov [7] is about 10 times faster than the proposed method, while our BOA-TFI is around three times faster than Jeong *et al.* [6].

We are working on an *ME* scheme that is tailored for the proposed method, which should make the ME both faster and more suited for the BOA-TFI scheme. Furthermore, in existing video codecs, the motion has to be (re-)estimated at the decoder for TFI purposes. This is in stark contrast to an HSVC scheme such as the one proposed in [13], which employs estimated "physical" motion, which does *not* have to be *(re-)estimated* at the decoder for TFI purposes; this significantly reduces the processing time of the proposed TFI framework.

As mentioned before, the focus of this paper is on the motion inference process, which is part of the *FI*. Most of the FI time is spent on mapping triangles from one frame to another in order to change *invert* and *infer* motion fields. In the current implementation, we map triangles of size 1×1; in regions away from moving objects, where motion is expected to be smooth, the triangle size could be greatly increased without any significant loss in quality. Initial investigations on a small number of sequences show that triangle merging can result in roughly 40–50 times fewer triangles, and hence a significant drop in the processing time can be expected. A much more thorough investigation of the trade-off between larger triangle size and interpolation quality is left for future work.

C) Results on Sintel sequences

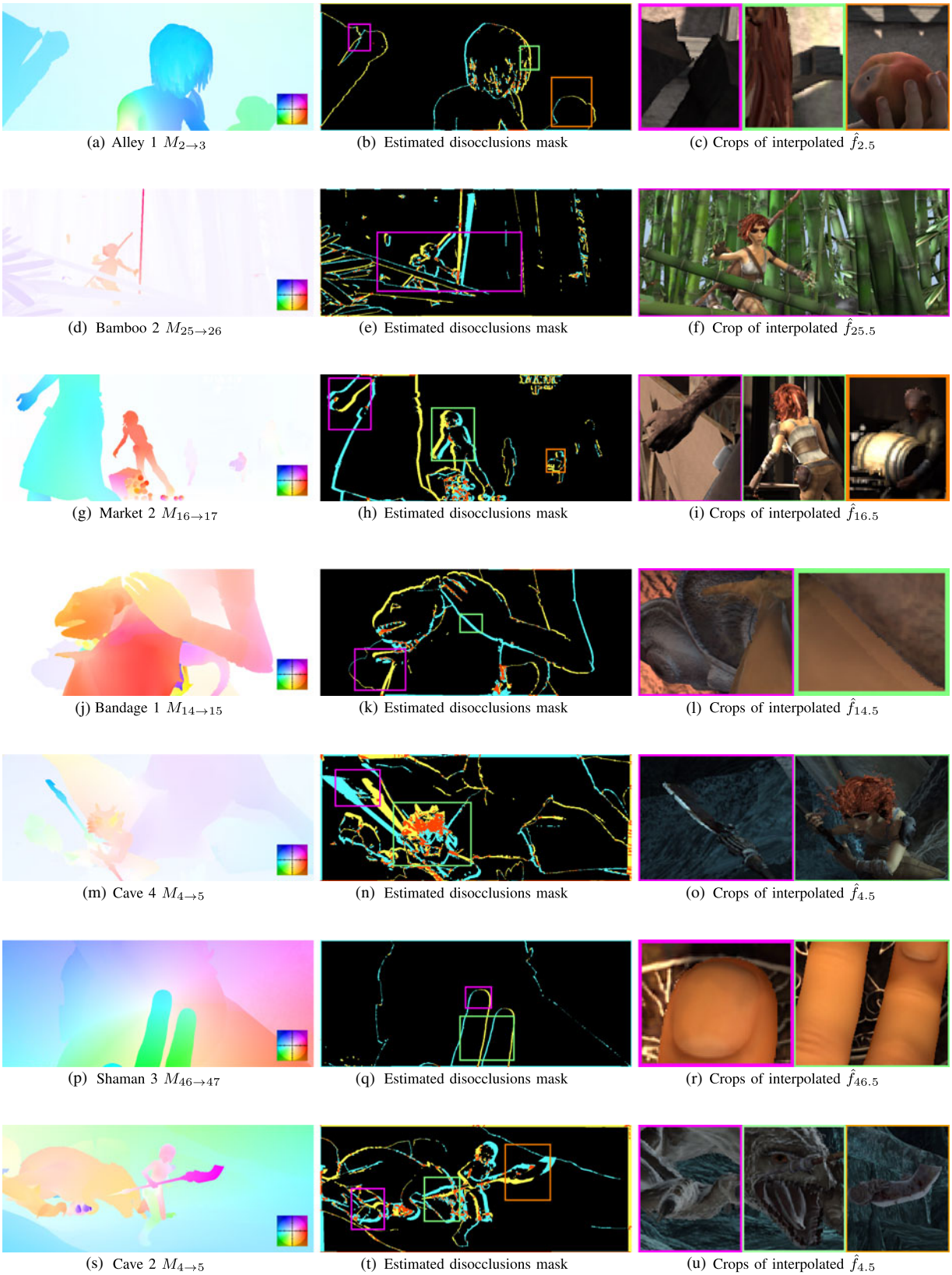
As mentioned earlier, the main focus of this work is on the motion inference process which produces geometrically consistent interpolated frames. For this to work, we need piecewise-smooth motion fields with sharp boundaries at moving object boundaries. The optical flow estimator we currently use to generate the results in Section VIII.A ([14]) is unidirectional, and hence has problems in finding the "correct" object boundary on the side of moving objects which do not have a correspondence; a parallel stream of work on bidirectional ME schemes is likely to provide further improved results.

To substantiate this claim and show what the proposed scheme is capable of if motion fields better suited for our TFI method are employed, we turn our attention to the Sintel sequence [19]; this computer-generated sequence is gaining a lot of popularity in the computer vision community because of its complexity. In order to show the performance of the scheme with "better" motion, we look at the quality of interpolated frames obtained using ground truth motion fields. Since both methods we compare ourselves to in Section VIII.A cannot make use of ground truth motion, we only show the results of our method, noting that any block-ME scheme would be highly challenged by the complexity of the underlying motion fields. Figure 12 shows sample interpolated frames generated by the proposed BOA-TFI method; full-resolution versions of the results, including animated versions, can be found on the website dedicated to this publication.[4]

The first column in the figure shows the (complex) ground truth motion fields, containing a variety of types of motion such as translation, rotation, zoom, and panning; furthermore, the motion magnitudes are much larger than on most natural sequences, resulting in larger regions of disocclusion around moving objects, as visualized in the second column of the figure. Because the ground truth motion fields for the Sintel sequence are only between adjacent frames, the frame we interpolate does not exist in the sequence, and hence we cannot compute a PSNR. As mentioned before, what ultimately counts is the perceived quality. One can see how the scheme is able to create high-quality reconstructed frames. The crops in the third row of Fig. 12 highlight difficult regions around moving object boundaries, where our BOA-TFI scheme switches from bidirectional to unidirectional prediction without smoothing the texture.

It is worth highlighting that the current scheme does not perform any texture optimization. In particular, the transition from uni- to bidirectional prediction can cause artifacts at the transition boundary if there are significant changes in illumination between the two reference frames. This can be observed in the right crop of the "Bandage 1" sequence Fig. 12l, most visible in the upper left part; the part of the wing which is brighter moves under the hand, and hence is only predicted from the left reference frame. The wing is significantly brighter in the left reference frame, and hence

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



(a) Alley 1 $M_{2\to3}$ (b) Estimated disocclusions mask (c) Crops of interpolated $\hat{f}_{2.5}$

(d) Bamboo 2 $M_{25\to26}$ (e) Estimated disocclusions mask (f) Crop of interpolated $\hat{f}_{25.5}$

(g) Market 2 $M_{16\to17}$ (h) Estimated disocclusions mask (i) Crops of interpolated $\hat{f}_{16.5}$

(j) Bandage 1 $M_{14\to15}$ (k) Estimated disocclusions mask (l) Crops of interpolated $\hat{f}_{14.5}$

(m) Cave 4 $M_{4\to5}$ (n) Estimated disocclusions mask (o) Crops of interpolated $\hat{f}_{4.5}$

(p) Shaman 3 $M_{46\to47}$ (q) Estimated disocclusions mask (r) Crops of interpolated $\hat{f}_{46.5}$

(s) Cave 2 $M_{4\to5}$ (t) Estimated disocclusions mask (u) Crops of interpolated $\hat{f}_{4.5}$

Fig. 12. TFI results on Sintel Sequence, which highlights the effectiveness of the proposed method to handle occluded regions. The first column shows the (color–coded) ground truth motion fields between the two reference frames, which, together with the two reference frames (not shown), form the input to our method in this experiment. The second column shows the *union* of the forward and backward disocclusion mask produced by the proposed BOA-TFI method, where *yellow* pixels are locations that get disoccluded between the previous and the interpolated frame; similarly, *cyan* are locations that get disoccluded between the future reference frame and the interpolated frame; *red* are regions that are not visible in either of the frames. The last column shows crops of the temporally interpolated frames obtained by the proposed BOA-TFI method.

the bidirectionally predicted part of the wing is darker than the unidirectionally predicted part. We plan to address this problem by looking into ways of optimizing the texture in such regions, which are easily identified from the disocclusion mask, and apply a selective filter in such transition regions. Even without any texture optimizations, we show that a good motion inference scheme is highly competitive with state-of-the-art TFI methods.

IX. CONCLUSIONS AND FUTURE WORK

This paper presents a TFI framework that creates geometrically consistent interpolated frames; explicit handling of occluded regions allows to resolve traditionally problematic regions around moving object boundaries. This is made possible by using high-quality *piecewise-smooth motion fields*, together with *motion discontinuities* at moving object boundaries. Motion discontinuities allow to reason about where foreground objects move, and enables to resolve double mappings, as well as assign reasonable motion in disoccluded regions.

We evaluate the method on a large set of natural and challenging computer-generated sequences, and our method compares favorably to state-of-the-art TFI methods. 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 development of a hierarchical ME scheme that is tailored to the proposed motion inference scheme. Furthermore, we plan to look into texture optimizations such as optical blur handling to further improve the visual quality of the upsampled frames.

ACKNOWLEDGEMENTS

We would like to thank Seong-Qyun Jeong [6] and Anton Veselov [7], for providing us with the results of their temporal frame interpolation methods.

REFERENCES

[1] Chan, S.H.; Nguyen, T.Q.: LCD motion blur: modeling, analysis, and algorithm. IEEE Trans. Image Process., **20** (8) (2011), 2352–2365.

[2] Girod, B.; Aaron, A.M.; Rane, S.; Rebollo-Monedero, D.: Distributed video coding. Proc. IEEE, **93** (1) (2005), 71–83.

[3] Choi, B.-D.; Han, J.-W.; Kim, C.-S.; Ko, S.-J.: Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. IEEE Trans. Circuits Systems Video Technol., **17** (4) (2007), 407–416.

[4] Wang, C.; Zhang, L.; He, Y.; Tan, Y.-P.: Frame rate up-conversion using trilateral filtering. IEEE Trans. Circuits Syst. Video Technol., **20** (6) (2010), 886–893.

[5] Dikbas, S.; Altunbasak, Y.: Novel true-motion estimation algorithm and its application to motion-compensated temporal frame interpolation. IEEE Trans. Image Proc., **22** (8) (2013), 2931–2945.

[6] Jeong, S.-G.; Lee, C.; Kim, C.-S.: Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization. IEEE Trans. Image Process., **22** (11) (2013), 4497–4509.

[7] Veselov, A.; Gilmutdinov, M.: Iterative hierarchical true motion estimation for temporal frame interpolation, in *IEEE Int. Workshop on Multimedia Signal Processing*, 2014, 1–6.

[8] Zhang, Y.; Xu, L.; Ji, X.; Dai, Q.: A polynomial approximation motion estimation model for motion compensated frame interpolation, IEEE Trans. Circ. Syst. Video Technol., 2015. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7118143

[9] Chin, Y.; Tsai, C.-J.: Dense true motion field compensation for video coding, in *Proc. IEEE Int. Conf. Image Processing*, 2013, 1958–1961.

[10] Kim, D.; Lim, H.; Park, H.: Iterative true motion estimation for motion-compensated frame interpolation. IEEE Trans. Circ. Syst. Video Technol., **23** (3) (2013), 445–454.

[11] Cho, Y.-h.; Lee, H.-y.; Park, D.-s. Temporal frame interpolation based on multiframe feature trajectory. IEEE Trans. Circ. Syst. Video Technol., **23** (12) (2013), 2105–2115.

[12] Rüfenacht, D.; Mathew, R.; Taubman, D.: Bidirectional, occlusion-aware temporal frame interpolation in a highly scalable video setting, in *Pict. Cod. Symp.*, 2015, 5–9.

[13] Rüfenacht, D.; Mathew, R.; Taubman, D.: A novel motion field anchoring paradigm for highly scalable wavelet-based video coding. IEEE Trans. Image Process., **25** (1) (2016), 39–52.

[14] Xu, L.; Jia, J.; Matsushita, Y.: Motion detail preserving optical flow estimation, IEEE Trans. Pattern Anal. Mach. Intell., **34** (9) (2012), 1744–1757.

[15] Wulff, J.; Black, M.J.: Modeling blurred video with layers, in *Eur. Conf. on Computer Vision*, vol. **8694**, 2014, 236–252.

[16] Mathew, R.; Taubman, D.; Zanuttigh, P.: Scalable coding of depth maps with R-D optimized embedding. IEEE Trans. Image Process., **22** (5) (2013), 1982–1995.

[17] Rüfenacht, D.; Mathew, R.; Taubman, D.: Bidirectional hierarchical anchoring of motion fields for scalable video coding, in *IEEE Int. Workshop on Multimedia Signal Processing*, September 2014, 1–6.

[18] Rüfenacht, D.; Mathew, R.; Taubman, D.: Hierarchical anchoring of motion fields for fully scalable video coding, in *Proc. IEEE Int. Conf. Image Processing*, October 2014, 3180–3184.

[19] Butler, D.J.; Wulff, J.; Stanley, G.B.; Black, M.J.: A naturalistic open source movie for optical flow evaluation, in *European Conf. Computer Vision*, October 2012, 611–625.

Dominic Rüfenacht received his B.Sc. in Communication Systems, in 2009, and the M.Sc. in Communication Systems with specialization in "Signals, Images and Interfaces" in 2011, both from the Swiss Federal Institute of Technology in Lausanne (EPFL). During his undergraduate studies, he was an exchange student at the University of Waterloo, Ontario, Canada, and did his Master's thesis at Philips Consumer Lifestyle in Eindhoven, Netherlands, entitled "Stereoscopic High Dynamic Range Video". From 2011 to 2013, he was with the Image and Visual Representation Group (IVRG) at EPFL as a Research Engineer, where he was working on computational photography problems, with emphasis on color and near-infrared imaging. He is currently pursuing a Ph.D. in Electrical Engineering at the University of New South Wales

(UNSW), Sydney, Australia. His research interests are both in computational photography and highly scalable image and video compression.

Reji Mathew received the B.E. degree from the University of Western Australia, Perth, Australia, in 1990, and the M.E. and Ph.D. degrees from the University of New South Wales (UNSW), Australia, in 1996 and 2010, respectively. He is currently with UNSW where he pursues his research interests in image and video coding, motion estimation, and scalable representations of motion and depth data. Reji's prior work experience includes employment with UNSW, Canberra (ADFA), from 1996 to 1997, Motorola Labs, Motorola Australian Research Centre, Sydney, from 1997 to 2003, and National ICT Australia, Sydney, from 2004 to 2005.

David Taubman received B.S. and B.E. (Electrical) degrees in 1986 and 1988 from the University of Sydney, and M.S. and Ph.D. degrees in 1992 and 1994 from the University of California at Berkeley. From 1994 to 1998 he worked at Hewlett-Packard's Research Laboratories in Palo Alto, California, joining the University of New South Wales in 1998, where he is a Professor in the School of Electrical Engineering and Telecommunications. Dr. Taubman is author with M. Marcellin of the book, "JPEG2000: Image compression fundamentals, standards and practice". His research interests include highly scalable image and video compression, motion estimation and modeling, inverse problems in imaging, perceptual modeling, and multimedia distribution systems. Dr. Taubman was awarded the University Medal from the University of Sydney. He has received two Best Paper awards: from the IEEE Circuits and Systems Society for the 1996 paper, "A Common Framework for Rate and Distortion Based Scaling of Highly Scalable Compressed Video"; and from the IEEE Signal Processing Society for the 2000 paper, "High Performance Scalable Image Compression with EBCOT".