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

Dominic Rüfenacht, Student Member, IEEE, Reji Mathew, Member, IEEE, David Taubman, Fellow, IEEE

Abstract—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 favourably to state-of-the-art TFI methods.

Index Terms—Temporal Frame Interpolation, Temporal Frame Upsampling, Geometrically Consistent Prediction, Occlusion-Handling, Motion Discontinuity Modelling, Bidirectional Hierarchical Anchoring of Motion Fields.

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 artefacts and motion blur in liquid crystal displays (LCDs) [1], or in distributed video coding, where temporally interpolated frames are used as side information for the Wyner-Zyv decoding [2]. In scalable video coding, where video can be decoded at different quality levels in terms of spatial, bit-rate, and temporal resolution, temporal frame interpolation is desirable when all information at a certain temporal level is quantized to zero.

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 methods have been proposed, which differ in terms of type of motion estimation 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 artefacts, various techniques which smooth the interpolated frames are employed. Choi et al. [3] use bilateral motion estimation, and block artefacts 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. Jeong et al. [5] perform motion-compensated frame interpolation using a multi-hypothesis motion estimation. The best motion hypothesis is selected by optimizing the cost function of a labelling problem. Pixels in the target frame are computed as a weighted combination of several pixels from the reference frame. They show improved reconstruction quality, at the expense of a significant increase in computational complexity. Veselov and Gilmudinov [6] propose a hierarchical bidirectional multi-stage motion estimation algorithm. They partition the target frame into non-overlapping, hierarchical blocks, and approximate the “true” motion flow. Each pixel is blended from multiple reference pixels. To avoid artificial boundaries created by block motion fields, Chin and Tsai [7] estimate a dense motion field, and apply the motion to each pixel location. Simple heuristics are used to handle holes and multiple mapped locations in the upsampled frame. Several frame interpolation methods have been proposed which try to detect occluded regions, and show improved performances compared to methods without occlusion handling. Kim et al. [8] use linearity checking between the estimated forward and backward motion fields to detect occluded regions. Cho et al. [9] use a bidirectional motion estimation scheme that is based on feature trajectory tracking, which allows to detect occluded regions.

In [10], we have shown how the 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 OBMC, as employed in most state-of-the-art schemes; such approaches can rarely be justified as modelling an underlying physical process, and often result in oversmooth interpolated frames.

This motion centric approach is well adapted to scalable compression schemes [11], 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 seamless 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 frame interpolation 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. [12] propose a motion detail preserving optical flow algorithm, which encourages sharp discontinuities in the motion field. Wulff and Black [13] 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 [10], 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 to 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 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 frame interpolation, 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 artefacts.

II. Notation

For the remainder of this paper, we denote the two reference frames as \( f_a \) and \( f_c \), respectively, and the frame to be interpolated (target) as \( f_b \). We further use \( M_{i \rightarrow j} \) to denote a motion field “anchored” at frame \( f_i \) and pointing to frame \( f_j \), such that each pixel in frame \( f_i \) is associated with a location in frame \( f_j \). \( B_i \) is used to denote motion discontinuity information (represented using breakpoints) anchored at frame \( f_i \). We use \( S_{i \rightarrow j} \) to denote a disocclusion mask anchored at frame \( f_i \), which is non-zero at all locations that are not visible (i.e., not predictable) from frame \( f_j \). Lastly, we use \( \hat{A} \) to denote an estimate of \( A \); for example, \( \hat{M}_{a \rightarrow b} \) in Fig. 1 denotes an estimated motion field anchored at frame \( f_a \) and pointing to frame \( f_b \).

III. Overview

Fig. 1 gives an overview of the proposed temporal frame interpolation method. Inputs to the method are the reference frames \( f_a \) and \( f_c \), and the motion field between them, \( M_{a \rightarrow c} \).1 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 work comes out of a highly scalable video coder [11], we use breakpoints to represent motion discontinuities [14]; breakpoints are very useful in a scalable video coder because of their high scalability attributes both in quality and resolution. See Sect. V 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 \( \hat{M}_{a \rightarrow b} \) of the motion field between frame \( f_a \) and the target frame \( f_b \) by scaling the parent motion field \( M_{a \rightarrow c} \) by a factor of 0.5. Next, we infer a motion field \( \hat{M}_{c \rightarrow b} \), which is anchored at frame \( f_c \) and pointing backwards to frame \( f_b \); to infer \( \hat{M}_{c \rightarrow b} \), both its temporal parent motion \( M_{a \rightarrow c} \), and its temporal sibling \( \hat{M}_{a \rightarrow b} \), are used.

The next step is to invert both \( \hat{M}_{a \rightarrow b} \) and \( \hat{M}_{c \rightarrow b} \), so that we obtain the two motion fields \( \hat{M}_{a \rightarrow b} \) and \( \hat{M}_{b \rightarrow c} \), 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

1We describe a way of estimating piecewise-smooth motion fields suitable for this work in Sect. VIII.
Hierarchical Anchoring

Estimated Discontinuity Information “Breakpoints”

Input Motion Fields

M_{c→e}

M_{a→c}

BP Est

Warp

Infer

Estimated Disocclusion Mask

Output Interpolated frame

Input Frames

M_{b→a}

M_{b→c}

S_{b→a}

S_{a→c}

S_{b→c}

M_{b→a}→

M_{b→c}→

M_{c→b}→

M_{a→b}→

M_{a→c}→

αM_{a→c}

0.5

M_{a→b}

M_{c→b}

M_{b→a}

M_{b→c}

Fig. 1. Overview of the proposed temporal frame interpolation method: The input to the scheme are a motion field $M_{a→c}$, as well as breakpoint fields estimated on $M_{a→c}$ for frame $f_a$, and on $M_{c→e}$ (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, $M_{a→b}$ is obtained by halving its parent motion field $M_{a→c}$, $M_{a→e}$, and $M_{a→b}$ are then used to infer the motion field $M_{c→b}$. The last step consists of inverting $M_{a→b}$ and $M_{c→b}$ to obtain $M_{b→a}$ and $M_{b→c}$. During the motion inversion process, we compute disocclusion masks $S_{b→a}$ and $S_{b→c}$, which are used to guide the bidirectional motion-compensated temporal frame interpolation (MCTFI) process to temporally interpolate the frame $f_b$. Breakpoints are used to resolve double mappings and handle occluded regions during both the motion inference and inversion process.

(a) Traditional Anchoring

(b) Hierarchical Anchoring

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

in the disocclusion masks $S_{b→a}$ and $S_{b→c}$, and are used to guide the bidirectional, occlusion-aware motion-compensated temporal frame interpolation (MCTFI) process.

IV. BIDIRECTIONAL HIERARCHICAL ANCHORING OF MOTION FIELDS

All current state-of-the-art video codecs anchor motion fields at the target frames. In [11], 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 frame interpolation, 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. Fig. 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, $M_{a→c}$ 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, $M_{a→c}$ is a block-based prediction field that minimizes the prediction residual, and is not reflective of “true motion”. As a result, $M_{a→c}$ cannot be scaled to point to the intermediate frame $f_b$, and hence has to be (re)estimated at the decoder. In our scalable video coding scheme, we closely model “true” motion fields, which can be scaled and hence readily be used to perform frame interpolation at the decoder.

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

One could be tempted to estimate $M_{c→a}$, and then compute $M_{c→b}$ as a scaled version of $M_{c→a}$. We avoid this strategy for two main reasons:

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

We instead infer $\hat{M}_{c→b}$, anchored at frame $f_c$, from the forward pointing motion field $M_{a→c}$ and its scaled version.
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 [14]. 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.

2In 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.

\[ M_{a\rightarrow b} = M_{a\rightarrow c} \circ (M_{a\rightarrow c})^{-1}. \] (1)

The fact that \( M_{c\rightarrow b} \) is completely defined by \( M_{a\rightarrow c} \) and \( M_{a\rightarrow b} \) has the key advantage that \( M_{c\rightarrow b} \) always “follows” \( M_{a\rightarrow b} \), 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.

V. 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 Sect. VI-A), as well as to extrapolate motion in disoccluded regions during the motion field inference process to obtain \( \hat{M}_{c\rightarrow b} \) (see Sect. VI-B). As this work builds upon a highly scalable video coding framework, we use a highly scalable way of coding discontinuities using breakpoints\(^2\); technical details on how breakpoints are estimated can be found in [14]. 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.

A. Inducing Motion Discontinuities from Breakpoints

For temporal frame interpolation, motion discontinuity information is not available for frame \( f_b \). In this work, we transfer such discontinuity information from the reference frames to the target frame using a hierarchical extension of the breakpoint warping scheme proposed in [15]. 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. Fig. 5 illustrates the three main steps of the proposed hierarchical temporal breakpoint induction method:

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

In cases where a warped line segment intersects an arc that already contains a spatially induced breakpoint, the temporally
The proposed temporal induction process consists of three steps at each resolution level \( \eta \): (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.

induced breakpoint always overwrites the spatially induced one.

The advantage of this hierarchical extension is that the temporal inducing constraints are tightest at the finest spatial resolution; spatially induced discontinuity information from coarser spatial levels can help completing discontinuity information in regions that are not compatible at finer spatial resolutions.

VI. 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 frame interpolation methods map either pixels or whole blocks from reference to target frames, which creates a variety of unwanted artefacts such as holes within objects because 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 piece-wise smooth motion, and employ a cellular affine warping (CAW) procedure first proposed in [15] 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 \( 1 \times 1 \) pixel, so that there are approximately twice as many triangles as there are pixels in the frame. 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.

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

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 \( x_j \) in the target frame \( f_j \). In other words, there are two locations \( x_{i,1} \) and \( x_{i,2} \) in \( f_i \), which are mapped by \( M_{i\rightarrow j} \) 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 \( x_{i,1} \) and \( x_{i,2} \)
in the reference frame $f_i$ as $l$; 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 $l$ which is on the same side as $x_{i,1}$; similarly, let $B^+$ denote the location on $l$ that is on the side of $x_{i,2}$. Because the motion discontinuity moves with the foreground object, either $y_{j,B^-} = M_{i\rightarrow j}(B^-)$ or $y_{j,B^+} = M_{i\rightarrow j}(B^+)$ 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, $y_{j,B^-}$ gets mapped onto motion discontinuities; therefore, $\hat{M}_{a\rightarrow b}(x_{i,1})$ gets recorded as foreground motion at location $x_j$ where the double mapping occurred (e.g., $\hat{M}_{b\rightarrow a}(x_j) = -\hat{M}_{a\rightarrow b}(x_{i,1})$).

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

$$S_{j\rightarrow i}(x) = \begin{cases} 1 & \text{x disoccluded} \\ 0 & \text{otherwise} \end{cases}.$$  

(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 $M_{a\rightarrow b}$, which we denote $S_{b\rightarrow a}$, and the other $S_{b\rightarrow c}$, obtained during the inversion of $M_{c\rightarrow b}$. They are used to generate the interpolated frame as explained in Sect. VII.

B. Motion Field Inference

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

Both $\hat{M}_{a\rightarrow b}$ and $\hat{M}_{c\rightarrow b}$ should reflect “true” motion with sharp discontinuities. In particular, $\hat{M}_{c\rightarrow b}$ 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 $M_{a\rightarrow c}$; 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 $\hat{M}_{c\rightarrow b}$ 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 $\hat{M}_{c\rightarrow b}$, 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. 8c, $D_1$ 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 $e_3$, formed by connecting $D_1'$ and $D_3'$, of the horizontal component ($v_{x}$) of the motion in Fig. 8d; the dashed blue line shows the motion assigned by the CAV 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.

VII. MOTION-COMPENSATED TEMPORAL FRAME INTERPOLATION

The last step is to interpolate the target frame $\hat{f}_b$. We use $W_{\hat{M}_{a\rightarrow c}} (f_j)$ to denote the warping process of frame $f_j$ to frame $\hat{f}_1$. The warping of frame $f_j$ to frame $f_1$, evaluated at location $x$, is then denoted as $f_{j\rightarrow i} (x) = (W_{\hat{M}_{a\rightarrow j}} (f_j))(x)$. Every pixel location $\hat{f}_b (x)$ in $\hat{f}_b$ is computed using $\hat{M}_{a\rightarrow a}$ and $\hat{M}_{b\rightarrow c}$, together with the estimated disocclusion maps $S_{b\rightarrow a}$ and $S_{b\rightarrow c}$, as:

$$\hat{f}_b (x) = \begin{cases} S_{b\rightarrow a} (x) f_{a\rightarrow b} (x) + S_{b\rightarrow c} (x) f_{c\rightarrow b} (x) & \kappa (x) > 0 \\ 0.5 (f_{a\rightarrow b} (x) + f_{c\rightarrow b} (x)) & \kappa (x) = 0 \end{cases}$$

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

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

VIII. 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 [12] motion detail preserving optical flow algorithm provides motion fields of sufficient quality to work with our proposed framework. The parent motion field $\hat{M}_{a\rightarrow c}$ is estimated using the default parameters of their implementation.

Next, we run Mathew et al.’s [14] breakpoint estimation scheme to estimate motion discontinuities on $\hat{M}_{a\rightarrow c}$. Fig. 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 [11], 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.

IX. EXPERIMENTAL EVALUATION AND DISCUSSION

In our previous work [10], 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 enhance the evaluation of the proposed method both qualitatively and quantitatively on various natural sequences (Sect. IX-A), and compare our performance to two state-of-the-art TFI methods ([6], [5]).

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 [16]; in Sect. IX-B, we further show qualitative results of the proposed motion inference scheme on various scenes from the Sintel sequence, which contain much larger amounts of disoccluded regions than the natural sequences.

### A. Results on Natural Sequences

In this section, we show results obtained on common test sequences, and compare them to two state-of-the-art temporal frame interpolation methods: [5] is a sophisticated multi-hypothesis testing framework, where a lot of effort is spent on texture optimization. [6] focuses 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 test sequences with a large variety of motion and texture complexity. 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 I presents the per sequence results, averaged over the 10 frames.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cactus</td>
<td>007-025</td>
<td>33.15 (+0.52)</td>
<td>31.28 (+2.39)</td>
<td>33.68</td>
</tr>
<tr>
<td>Kimono</td>
<td>001-019</td>
<td>33.92 (-0.68)</td>
<td>33.39 (-1.14)</td>
<td>33.25</td>
</tr>
<tr>
<td>Kimono</td>
<td>175-193</td>
<td>39.94 (+0.83)</td>
<td>40.14 (+0.63)</td>
<td>40.77</td>
</tr>
<tr>
<td>Rushhour</td>
<td>049-067</td>
<td>35.16 (-0.52)</td>
<td>34.91 (-0.27)</td>
<td>34.64</td>
</tr>
<tr>
<td>Shields</td>
<td>101-119</td>
<td>35.90 (+0.16)</td>
<td>35.06 (+0.99)</td>
<td>36.05</td>
</tr>
<tr>
<td>Shields</td>
<td>385-403</td>
<td>33.87 (+3.91)</td>
<td>35.55 (+2.22)</td>
<td>37.77</td>
</tr>
<tr>
<td>Stockholm</td>
<td>261-279</td>
<td>36.58 (+1.26)</td>
<td>37.09 (+0.75)</td>
<td>37.84</td>
</tr>
<tr>
<td>Park</td>
<td>139-157</td>
<td>38.26 (+0.98)</td>
<td>38.81 (+0.43)</td>
<td>39.24</td>
</tr>
<tr>
<td>Parkrun</td>
<td>145-163</td>
<td>30.62 (+1.20)</td>
<td>30.95 (+0.87)</td>
<td>31.32</td>
</tr>
<tr>
<td>Station2</td>
<td>021-039</td>
<td>40.05 (+1.79)</td>
<td>40.91 (+0.93)</td>
<td>41.84</td>
</tr>
<tr>
<td>Mobcal</td>
<td>361-379</td>
<td>29.13 (+8.59)</td>
<td>34.75 (+2.98)</td>
<td>37.73</td>
</tr>
<tr>
<td>Terrace</td>
<td>139-157</td>
<td>33.27 (+4.36)</td>
<td>34.20 (+3.43)</td>
<td>37.63</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>34.99 (+1.87)</td>
<td>35.59 (+1.27)</td>
<td>36.85</td>
</tr>
</tbody>
</table>

While reporting average 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 of global motion. 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, 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.

B. 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 Sect. IX-A ([12]) is unidirectional, and hence has problems on the side of moving objects which do not have a correspondence; a parallel stream of work on bidirectional motion estimation 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 [16]; 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 Sect. IX-A cannot make use of ground truth motion, we only show results of our method, noting that any block-motion estimation scheme would be highly challenged by the complex motion fields. Figure 12 shows sample interpolated frames generated by the proposed BOA-TFI method.

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 larger than on most natural sequences, resulting in larger regions of disocclusion around moving objects. Because the ground truth motion fields for the Sintel sequence are only between adjacent frames, and hence the frame we interpolate does not exist in the sequence; this means that 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 artefacts at the transition boundary if there are significant changes in illumination between the two reference frames. This can be seen in the right crop of the “Bandage 1” sequence Fig. 12f, 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 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.

X. CONCLUSIONS AND FUTURE WORK

This paper presents a temporal frame interpolation (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 favourably 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 motion estimation 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.

ACKNOWLEDGMENTS

We would like to thank Seong-Qyun Jeong [5] and Anton Veselov [6], for providing us with the results of their temporal frame interpolation methods.

REFERENCES