IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 September 2023, accepted 6 October 2023, date of publication 11 October 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323873

RESEARCH ARTICLE

Learned Wavelet Video Coding Using Motion Compensated Temporal Filtering

ANNA MEYER, (Graduate Student Member, IEEE), FABIAN BRAND, (Graduate Student Member, IEEE), AND ANDRÉ KAUP, (Fellow, IEEE)

Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany

Corresponding author: Anna Meyer (anna.meyer@fau.de)

This work was supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Project 461649014.

ABSTRACT This paper presents an end-to-end trainable wavelet video coder based on motion-compensated temporal filtering. Thereby, it introduces a different coding scheme for learned video compression, which is dominated by residual and conditional coding approaches. By performing discrete wavelet transforms in temporal, horizontal, and vertical dimensions, an explainable framework with spatial and temporal scalability is obtained. This paper investigates a novel trainable motion-compensated temporal filtering module implemented using the lifting scheme. It demonstrates how multiple temporal decomposition levels can be considered during training. Furthermore, larger temporal displacements owing to the coding order are addressed and an extension adapting to different motion strengths during inference is introduced. The experimental analysis compares the proposed approach to learning-based coders and traditional hybrid video coding. Especially at high rates, the approach exhibits promising rate-distortion performance. The proposed method achieves average Bjøntegaard Delta savings of up to 21% over HEVC, and outperforms state-of-the-art learned video coders.

INDEX TERMS Convolutional neural networks, deep learning, discrete wavelet transforms, motion compensation, motion estimation, scalability, video codecs, video coding, video compression, video signal processing.

I. INTRODUCTION

Following the progress of learned image compression, there have been significant advances in learned video compression. Built on learned image coders, video coding approaches exploit temporal redundancies by following two main paradigms: residual and conditional coding. Residual coders [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] largely take over the structure of known hybrid video coders such as VVC [11]. Using motion-compensated inter prediction, the residual between the predicted and current frame is compressed and then transmitted. Instead of transmitting a difference signal, conditional coders compress the current frame directly under the condition that both the encoder and decoder know the prediction. Since the introduction of conditional coding in learned video compression by a framework called DCVC [12], there have been several improvements of DCVC [13], [14] as well as other conditional coding schemes based on a generative model [15] or transformers [16], [17]. With these developments, conditional coding currently outperforms residual coders and represents the state of the art in learned video coding.

This paper investigates a different coding scheme visualized in Fig. 1: learned wavelet video coding. It performs a discrete wavelet transform (DWT) in temporal, horizontal, and vertical dimensions. Specifically, an end-to-end trainable wavelet video coder based on Motion-Compensated Temporal Filtering (MCTF) [18] is introduced. Traditional MCTF as proposed by Ohm [18] and improved by Choi and Woods [19], incorporates motion compensation into the temporal wavelet transform. Until the early 2000s, MCTF-based wavelet video coding was an active research topic as a

The associate editor coordinating the review of this manuscript and approving it for publication was Gerard-Andre Capolino.

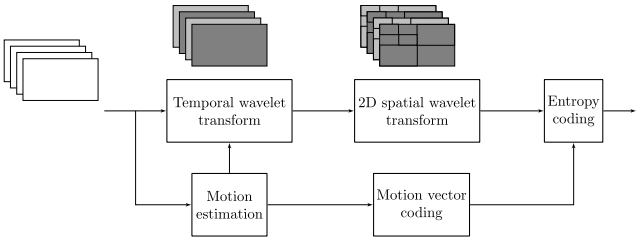

FIGURE 1. Schematic overview of the wavelet video coding scheme. This paper introduces a novel trainable version of the coding scheme. A temporal wavelet transform is followed by a 2D wavelet transform in horizontal and vertical dimensions. By incorporating motion compensation into the temporal wavelet transform, Motion-Compensated Temporal Filtering (MCTF) is performed.

scalable alternative to predictive transform coders. With the success of the video coding standard H.264/AVC [20], hybrid video coding approaches have dominated the field. Transform coding has emerged as the predominant principle in learned image and video compression. Here, the foundation of most popular coders [21], [22], [23] is based on nonlinear transform coding [24].

Recently, employing a learned spatial wavelet transform for end-to-end image compression has shown great potential by achieving state-of-the-art performance [25]. Motivated by this emerging topic of trainable wavelet transforms for compression, the novel learned MCTF video coding approach is built on top of the wavelet image coder called iWave++ [25]. The MCTF video coder provides a flexible framework that supports lossless compression. In addition, MCTF enables a fully scalable video coder. Compared to other learned approaches, which usually do not support input in YUV 4:2:0 format, wavelet video coding allows arbitrary input formats.

The focus of this paper is on the investigating a novel trainable MCTF module and compressing the obtained temporal subbands with the state-of-the-art wavelet image coder iWave++ [25]. The contributions of this paper are as follows:

- Introduction of the first end-to-end trainable wavelet video coding scheme. To date, there have been no learned video compression approaches based on MCTF.
- Presentation of a training strategy for multiple temporal decomposition levels in MCTF.
- Investigation of large temporal displacements due to the MCTF coding structure and a first solution for handling these cases more efficiently.
- Proposal of a content-adaptive MCTF approach that adapts to different types of motion during inference.

II. STATE OF THE ART

A. LEARNED VIDEO COMPRESSION

DVC [1], [2] was the first learning-based deep video compression framework. It follows the structure of a traditional hybrid P-frame codec but replaces its modules for motion estimation, motion vector and residual compression by neural networks. With the feature-space video coding network FVC [4], the DVC framework was significantly improved by performing these operations in the feature space. The coarse-to-fine framework C2F [5] further advanced

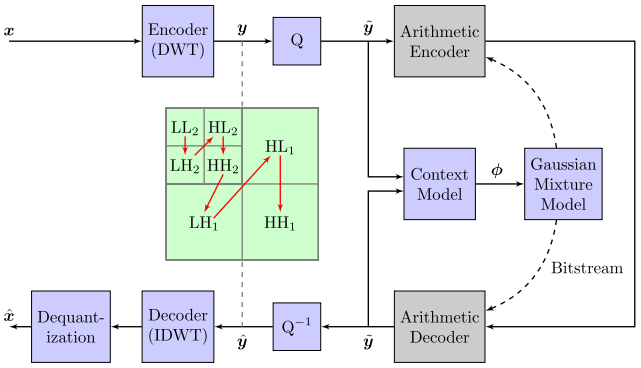

FIGURE 2. Overview of the end-to-end image compression method iWave++ [25]. x is a single luma or chroma channel of an image in the YCbCr color space. The red arrows indicate the coding order of the subbands y. Trainable modules are colored in blue. For visualization, the subbands of two decomposition levels are shown.

residual coding using two-stage motion compensation at different resolutions and mode prediction networks.

Conditional coding can offer theoretical benefits over residual coding [26] and learning-based frameworks allow for its straightforward implementation via conditional autoencoders [27], [28]. The DCVC [12] framework has attracted greater attention to conditional coding for learned video compression. Conditioning on the temporal context in the feature space [13], and an extended entropy model with a latent prior in addition to quantization at different granularities [14] made the DCVC framework reach state-of-the-art performance. Another conditional coding approach [17] follows the structure of DCVC but uses a transformer-based entropy model. There are also frameworks based on augmented normalizing flows [15] or without an explicit motion model, such as the video compression transformer VCT [16].

B. WAVELETS FOR LEARNED IMAGE COMPRESSION

The traditional discrete wavelet transform has desirable properties for image and video coding. Its compromise between spatial and frequency resolution fits the correlation structure of image data: edges can be coded more efficiently in the spatial domain, whereas smooth shades and regular textures can be better modeled in the frequency domain. Hence, the image compression standard JPEG2000 [29] and the Dirac video coder [30] employ a DWT as an alternative to the discrete cosine transform. The coders rely on the lifting scheme [31] for a fast and efficient implementation of the DWT. With the lifting structure, the DWT can be performed in place by factoring its calculation into multiple lifting steps. At the same time, the lifting structure allows the construction of new wavelet filters, so-called second-generation wavelets. Moreover, the lifting scheme is a reversible structure and is thus well suited for realizing lossless transforms that can be incorporated into learning-based frameworks.

Convolutional Neural Networks (CNNs) allow the optimization of wavelet transforms based on a set of training images [32]. Such a learned wavelet transform implemented via the lifting scheme has been shown to outperform the

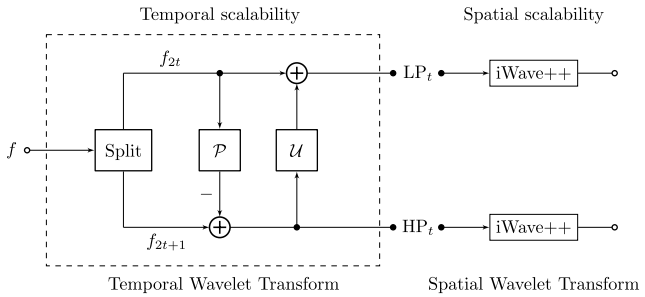

FIGURE 3. Overview of the proposed wavelet video coding scheme for one temporal decomposition level with two frames. f denotes the input video sequence.

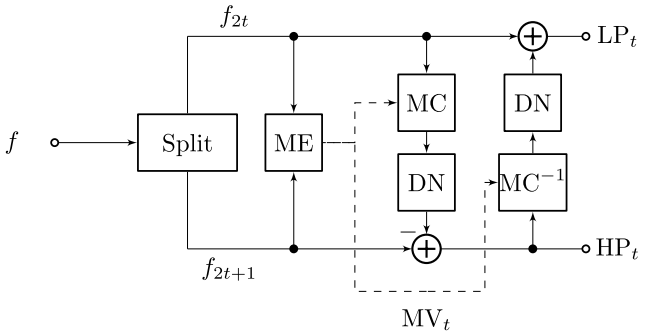

FIGURE 4. Details on prediction and update filters. f denotes the input video sequence. The "ME" module contains motion estimation and motion vector coding. Its output MV_t corresponds to the decoded motion vectors at time instance t. MC stands for motion compensation and MC^{-1} for inverse motion compensation. The "DN" modules represent residual CNN-based filter operations.

wavelet filters used by JPEG2000. The learned wavelet transform forms the basis of the end-to-end trainable wavelet image coder iWave++ [25]. An overview of iWave++ is shown in Fig. 2. First, the encoder performs a CNN-based DWT with four decomposition levels. The obtained tree-structured subbands constitute a hierarchical representation at different resolutions. For transmission, the wavelet coefficients are quantized using scalar quantization with a trainable parameter. Subsequently, a CNN-based context model estimates the entropy parameters of a Gaussian mixture model employed for adaptive arithmetic coding. The context model exploits correlations within the current subband to be coded and across subbands from different decomposition levels. After an inverse discrete wavelet transform (IDWT) is performed by the decoder, a post-processing module compensates for quantization artifacts.

Learned wavelet image compression provides a flexible framework. A 3D version of iWave++ [33] has been employed for lossless and lossy medical image compression, that is, for coding 3D volume data without temporal information. An extension through an affine wavelet transform module further improved volumetric image compression performance [34]. The low- and highpass subbands obtained from the lifting scheme are re-scaled by an affine map computed based on the output of the prediction and update filters.

Dong et al. [35] proposed a partly trainable wavelet video coder that follows a "t+2D" decomposition structure. First, they perform a temporal wavelet transform taken from [36]. Afterwards, they code the obtained temporal subbands using a trainable entropy parameter estimation module that largely takes over the structure of iWave++. In addition, Dong et al. enabled quality scalability via bitplane coding. This paper focuses on a trainable temporal wavelet transform instead to obtain a fully CNN-based wavelet video coder.

III. LEARNED WAVELET VIDEO CODING

In the following section, the end-to-end trainable wavelet video coding scheme is introduced. Fig. 3 provides an overview of the proposed approach. The temporal wavelet transform realized via MCTF provides temporal scalability. The obtained temporal low- and highpass subbands are coded using dedicated iWave++ [25] image compression models. Its spatial 2D wavelet transform yields spatial scalability.

First, the concept of wavelet video coding for one temporal decomposition level, that is, for coding two frames is explained. Subsequently, multiple temporal decomposition levels are discussed in Section IV.

A. TRAINABLE TEMPORAL WAVELET TRANSFORM

1) LIFTING SCHEME

The lifting structure [31] provides a flexible and efficient implementation of the DWT. The temporal lifting scheme illustrated in Fig. 3 consists of the three steps split, predict, and update. In the first step, the input video sequence f is split into even- and odd-indexed frames f_{2t} and f_{2t+1}. In the next step, the odd frames are predicted from the even frames with the prediction operator $\mathcal{P}$. A temporal highpass subband (HP_t) is obtained as $\mathrm{HP}_t = f_{2t+1} - \mathcal{P}(f_{2t})$. Subsequently, an update step is performed according to $\mathrm{LP}_t = f_{2t} + \mathcal{U}(\mathrm{HP}_t)$ resulting in a temporal lowpass subband (LP_t). The inverse lifting scheme is obtained by reversing the order of the operations and inverting the signs. Rounding the output of the prediction and update operators yields an integer-to-integer temporal DWT required for lossless reconstruction [37].

2) PREDICTION AND UPDATE FILTERS

Fig. 4 illustrates the detailed structure of the prediction and update filters. For the prediction step, motion estimation between the even and odd frames f_{2t} and f_{2t+1} is performed to obtain the motion vectors at time instance t. The motion vectors are employed for motion compensation, followed by a denoising filtering module (DN). In the update step, the motion vectors are inverted to perform inverse motion compensation (MC^{-1}) followed by another denoising module. Due to the update step, the even frame is effectively low-pass filtered along the motion trajectory. The temporal lowpass filtering separates noise from content over time.

Applying a denoising filter after forward and inverse motion compensation has been shown to improve compression efficiency in scalable lossless wavelet coding of dynamic CT data [38]. This paper follows the same processing order and structure but uses trainable denoising filters, allowing for

flexibility during training. The denoising filters have the same residual filter structure as the prediction and update filters of the CNN-based spatial DWT in iWave++ [25].

B. MOTION ESTIMATION AND MOTION VECTOR COMPRESSION

The approaches for motion estimation and motion vector coding follow the state-of-the-art learned video coder DCVC-HEM [14]. During motion estimation, a dense optical flow field is estimated using a Spatial Pyramid Network (SPyNet) [39]. With six pyramid levels, the input of SPyNet is 6× downsampled. At every pyramid level, a network computes the residual flow based on the upsampled flow from the preceding level, and thus deals with relatively small motion.

To code the motion vectors obtained from SPyNet, a motion vector encoder computes a 64-channel latent representation with a 16× downscaled spatial resolution. The latents are discretized using multi-granularity quantization. The entropy model uses a hyper prior and a dual spatial prior. The latter is a two-step coding approach that exploits channel redundancies, which allows parallelization, in contrast to an autoregressive prior. The latent prior employed by DCVC-HEM conditions the entropy model on previously coded motion vector latents and is omitted for the MCTF coder. Because training is performed using only two frames, as detailed in Section IV-B, only one motion vector latent is available. For more details on motion vector compression, please refer to [14].

IV. DYADIC TEMPORAL DECOMPOSITION

A dyadic decomposition [29] recursively applies a wavelet transform in the temporal direction to the lowpass of the previous decomposition stage. Thus, different temporal resolutions are obtained at each decomposition level for temporal scalability. With the dyadic decomposition structure, the number of frames contained in a group of pictures (GOP) is equal to powers of two. This paper investigates GOPs containing up to eight frames.

A. CODING ORDER AND TEMPORAL SCALABILITY

Fig. 5 illustrates the coding order of MCTF for a GOP consisting of 8 frames. Because MCTF is an open-loop structure, motion estimation is performed on the original frames instead of the decoded frames. In the first temporal decomposition level, the operator $\mathcal{P}_1$ predicts all odd-indexed frames from the respective preceding frame. The resulting four temporal highpass frames $h_{1,t}$ and their corresponding motion vectors can be directly coded. Next, the temporal lowpass frames are obtained from the update operation $\mathcal{U}_1$ which receives the highpass frames as input. After the first temporal decomposition level, there are four temporal lowpass frames. MCTF repeats this decomposition in the temporal direction until only the single temporal lowpass frame $l_{3,0}$ in decomposition level $j = 3$ is left. Overall, the highpass frames $h_{1,t}$ from the first decomposition level can be coded first, followed by the highpass frames from the deeper temporal levels.

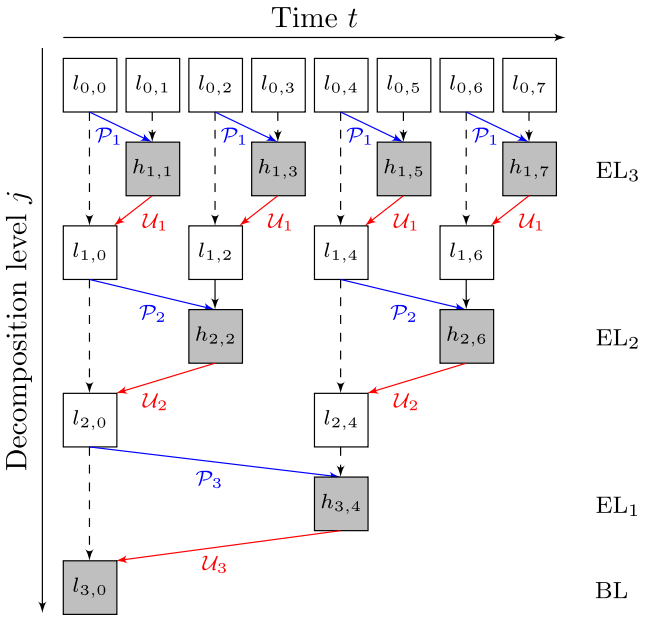

FIGURE 5. Coding order for a GOP size of 8. The temporal lowpass and highpass subbands are denoted as $l_{j,t}$ and $h_{j,t}$. The gray frames are coded from temporal decomposition level 1 to 3.

Finally, the lowpass frames $l_{3,0}$ from the lowest temporal decomposition level are transmitted.

Note that the distance between frames d in the temporal direction increases with every temporal decomposition level j according to $d = 2^{j-1}$. Hence, the frame distance d is equal to 4 in temporal decomposition level $j = 3$. This is disadvantageous in terms of rate-distortion performance compared with regular P frame coding with a frame distance of $d = 1$ for every P frame. However, MCTF has the benefit of providing temporal scalability: the lowpass subbands are similar to the original sequence and therefore correspond to a Base Layer (BL). The highpass subbands contain residual information that serves as an Enhancement Layer (EL). The further the input video sequence is decomposed in the temporal direction, the more ELs are available. For a GOP size of 8, there are three ELs as indicated in Fig. 5. Owing to the different temporal decomposition levels, dedicated MCTF filtering, motion estimation, and motion vector compression networks for each temporal decomposition level are beneficial. The benefits of the different MCTF stages are evaluated in Section V-B2.

On the decoder side, the inverse MCTF is performed by reversing the order of the prediction and update filters.

B. TRAINING STRATEGY AND LOSS

This paper adopts a multi-stage training strategy, of which Table 1 provides an overview. During the entire training procedure, each training sample consists of two frames. In the first part, a single MCTF stage is trained (training stage 1-3), and more stages are added in the second training part (training stage 4-5) depending on the GOP size. Thus, dedicated models for different GOP sizes are trained to consider the varying number of temporal decomposition levels. The two

TABLE 1. Training schedule for a GOP size of 4/8. A training sample consists of two frames in each training stage. LR denotes the learning rate, $d_{\max}$ the maximum frame distance between two frames in a training sample, and "parts" refers to the trainable components of the network. "All" parts include the MCTF stages and the iWave++ models.

MCTF stages	Parts	$d_{\max}$	Loss	LR	Epochs
Single	MCTF	1	D_{ME}	1×10^{-4}	1
Single	MCTF	1	$D_{\text{ME}} + R_{\text{MV}}$	1×10^{-4}	3
Single	All	1	$\mathcal{L}_{\text{full}}$	1×10^{-5}	5
Multiple (2/3)	MCTF	2/4	$\mathcal{L}_{\text{full}}$	5×10^{-5}	2
Multiple (2/3)	All	2/4	$\mathcal{L}_{\text{full}}$	1×10^{-5}	3

iWave++ models employed for coding the temporal lowpass and highpass subbands are initialized with models pretrained on image data.

1) FIRST TRAINING PART: SINGLE MCTF STAGE

During the first two training stages, only the network components for MCTF are trainable. They consist of motion estimation, motion vector compression, and DN modules. In the first stage, the loss is the distortion D_{ME} between the frame to be predicted f_{2t+1} and the prediction $\mathcal{P}_1(f_{2t})$. The second stage additionally considers the rate R_{MV} required for motion vector coding. In the next stage, the entire network is trainable and the loss is the regular rate-distortion loss.

The full rate-distortion loss for two frames reads:

$$\mathcal{L}_{\text{full}} = \sum_{i=0,1} R_{\text{all},i} + \lambda \cdot D_{\text{MSE}}(f_i, \hat{f}_i),$$

where i denotes the frame number and the distortion term corresponds to the Mean Squared Error (MSE) between the original frame f_i and the reconstructed frame $\hat{f}_i$. $R_{\text{all},i}$ consists of the rate required to code the temporal subbands using an iWave++ model. If the corresponding frame i is coded as a temporal highpass subband, $R_{\text{all},i}$ also includes R_{MV}. This paper considers lossy compression, where the only information loss stems from the scalar quantization operation of the iWave++ models.

2) SECOND TRAINING PART: MULTIPLE MCTF STAGES

To account for multiple temporal decomposition levels, multiple MCTF networks are used, where the additional MCTF stages are initialized with the parameters of the already available MCTF stage. For a GOP size of 4 with two temporal decomposition levels, two MCTF stages and a maximum frame distance $d_{\max}$ of two are used. For every batch element, a random frame distance between one and $d_{\max}$ is selected. Depending on the frame distance, a different MCTF stage with different networks is chosen. Thus, for a GOP size of 4, it is randomly alternated between optimizing the first MCTF stage with a frame distance of one and the second MCTF stage with a frame distance of two. Thereby, the different MCTF stages share the iWave++ models employed for coding the temporal lowpass and highpass subbands.

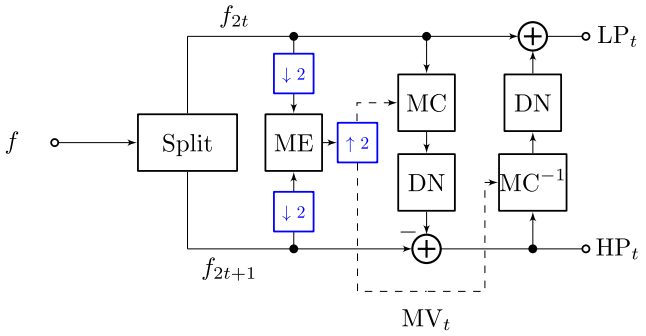

FIGURE 6. Lifting structure with downsampling strategy for decomposition levels $j > 1$ (*MCTF-DS*). Here, the "ME" module performs motion estimation on downscaled input frames. The reconstructed motion vectors MV_t obtained from motion vector compression are upscaled before being used for forward and inverse motion compensation.

In the last two training stages, again only the MCTF components are trained first and then all network modules are jointly optimized as can be seen in Table 1. To consider inverse MCTF for multiple decomposition levels during training, experiments were conducted using four frames per batch element. However, they showed that training becomes unstable, and the final rate-distortion performance is significantly worse than training with two frames and one temporal level.

For a GOP size of 8, the number of MCTF stages is increased from two to three. The maximum frame distance $d_{\max}$ in the last two stages (see Table 1) is set to 4 to account for the GOP structure shown in Fig. 5.

C. DOWNSAMPLING STRATEGY FOR TEMPORAL DISPLACEMENTS IN MCTF

The larger the temporal decomposition level, the larger the temporal distance between the frames in the original sequence (see Section IV-A). Therefore, considerably larger temporal displacements are possible. If the motion is too strong for the motion estimation network to predict accurately, prediction errors can lead to ghosting and error propagation across decomposition levels.

To address larger motion, computing and transmitting motion vectors at a lower spatial resolution for temporal decomposition levels larger than one is proposed. Specifically, the current frame and reference frame before motion estimation are downscaled by a factor of two for every temporal decomposition level $j > 1$ as illustrated in Fig. 6. Hence, the motion vectors are coded at lower resolution and upscaled after the motion vector decoder. Both bilinear down- and upsampling are performed. The upscaled motion vectors are then used for the forward and inverse MCTF. The proposed downsampling strategy (*MCTF-DS*) does not require additional training, and its benefits are evaluated in Section V-B2.

D. CONTENT-ADAPTIVE MCTF (MCTF-CA)

The coding efficiency of MCTF is highly dependent on the motion-compensated prediction quality as motion estimation errors propagate to higher temporal decomposition levels.

Even with the downsampling strategy, the motion present in a scene can be too strong for the motion estimation network or occluded regions can limit the prediction quality. Therefore, adaptive temporal scaling for each video sequence can lead to improved coding efficiency compared with uniform dyadic temporal decomposition by mitigating ghosting and thus error propagation. Lanz et al. [40] investigated content-adaptive wavelet lifting for scalable lossless coding of medical data by choosing the number of temporal decomposition levels based on the sequence content. This paper proposes the adoption of a content-adaptive wavelet lifting approach for our lossy wavelet video coder, which is referred to as *MCTF-CA*. This approach does not require additional training.

In the following section, the concept of content-adaptive MCTF for a GOP size of 8 is explained. During inference, the coding costs for a GOP consisting of 8 frames are optimized. As a cost criterion, the rate-distortion cost for $N = 8$ frames is evaluated as:

$$\mathcal{C}_N = \sum_{i=0}^{N-1} R_{\text{all},i} + \lambda \cdot D_{\text{MSE}}(f_i, \hat{f}_i),$$

where the tradeoff parameter λ is chosen according to the value employed for training the MCTF model. With one MCTF model trained for a GOP size of 8, evaluate different options for coding the current GOP. Subsequently, the variant with the minimum coding cost is chosen:

$$\min\left(\mathcal{C}_{8,\text{GOP8}},\ \mathcal{C}^{\text{DS}}_{8,\text{GOP8}},\ \mathcal{C}_{8,\text{GOP4}},\ \mathcal{C}^{\text{DS}}_{8,\text{GOP4}},\ \mathcal{C}_{8,\text{GOP2}}\right),$$

where the notation $\mathcal{C}^{\text{DS}}_{8,\text{GOP4}}$ denotes the cost of coding 8 frames in smaller GOPs of size 4 with the downsampling strategy, for example. Either a GOP size of 8 is coded or split into several smaller GOPs. Here, two GOPs of size 4 or four GOPs of size 2 are possible. In addition, it is decided whether to use the downsampling strategy or not.

In total, five options are considered for coding a GOP with 8 frames. The choice for each GOP needs to be transmitted to the decoder side. However, the overhead of transmitting three bits per eight frames is negligible. Hence, binary encoding is used to signal the content-adaptive choice for a coding unit with eight frames.

For a GOP size of 4, there are three options: Two GOPs of size 2 and one GOP of size 4, with or without downsampling.

V. EXPERIMENTS AND RESULTS

A. EXPERIMENTAL SETUP

1) TRAINING DETAILS

The networks described above are implemented using the PyTorch framework. The Vimeo90K data set [41] is used for training and the batch size is set to 8. During training, patches of size 128×128 are cropped from the luma channel of the respective training sample, whereas no cropping is performed during inference. By choosing the rate-distortion trade-off parameter according to $\lambda = \{0.007, 0.01, 0.03, 0.05, 0.08\}$, five models are obtained for each GOP size. AdamW [42] is used as optimizer. Furthermore, the iWave++ models pretrained on luma data from [43] are used for temporal subband coding. SPyNet [39] is initialized with the "sintel-final" model[1] trained on a synthetic data set.

As described in Section IV-B, separate models with multiple MCTF stages are trained for GOP sizes of 4 and 8, because the seven frames available in sequences from the Vimeo90K data set allow considering up to three temporal decomposition levels, that is, a maximum GOP size of 8. In line with the MCTF evaluation setup from Dong et al. [35] with a GOP size of 8, it is shown that in this setting, MCTF performs competitive to state-of-the-art coders.

2) TEST CONDITIONS

The UVG [44] and MCL-JCV [45] data sets are used for testing. The sequences in both data sets have a resolution of 1920×1080 and are in YUV 4:2:0 format. UVG consists of 7 sequences and MCL-JCV of 30. To consider a different resolution of 1280×720, the JCT-VC class E data set (HEVC E) containing three YUV 4:2:0 sequences are used. The test conditions in [14] are followed by evaluating on the first 96 frames of each sequence. In addition, the evaluation includes three sequences from the UVG 4K [44] data set (*CityAlley*, *FlowerFocus*, *FlowerKids*) with a resolution of 3840×2160 and testing is performed on the first 24 frames.

DCVC-HEM[2] [14] and DCVC[3] [12] are evaluated with GOP sizes of 4 and 8 for a fair comparison with the MCTF approach. Thereby, publicly available models from the authors are used, which were trained on Vimeo90K. As a traditional hybrid video coder, HM 16.25[4] is included. HM is used in the Lowdelay P (LD-P) configuration because the learned video coders only support unidirectional motion estimation. HM is evaluated in its default main profile with an intra period and GOP sizes of 4 and 8 as well.

The evaluation is performed in terms of RGB-PSNR, as this is common in learned video compression, and the aim of this paper is to provide comparable measurements. The MCTF approach and HM receive the input video sequence in YUV 4:2:0 format, whereas the input is converted to RGB 4:4:4, as required by DCVC-HEM and its predecessor DCVC. The wavelet video coder supports input data in YUV 4:2:0 format as well as in 4:4:4 format, because the color channels are coded independently by iWave++. The motion vectors are computed based on the luma channel. They are re-used for the chroma channels, and bilinear downsampling is performed if necessary.

B. EXPERIMENTAL RESULTS

1) COMPARISON TO STATE-OF-THE-ART VIDEO CODERS

a: RATE-DISTORTION CURVES

The novel approach is compapred to HM, the state-of-the-art learned video coder DCVC-HEM [14], and its predecessor

[1] https://github.com/sniklaus/pytorch-spynet
[2] https://github.com/microsoft/DCVC/tree/main/DCVC-HEM
[3] https://github.com/microsoft/DCVC/tree/main/DCVC
[4] https://vcgit.hhi.fraunhofer.de/jvet/HM/-/releases/HM-16.25

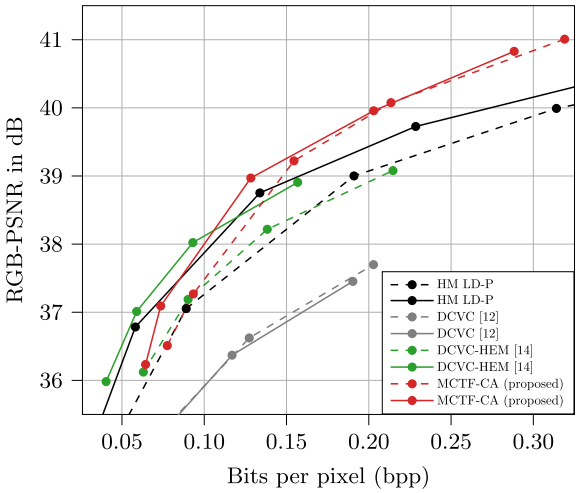

FIGURE 7. Rate-distortion evaluation on the UVG data set. Solid lines correspond to a GOP size of 8 and dashed lines to a GOP size of 4.

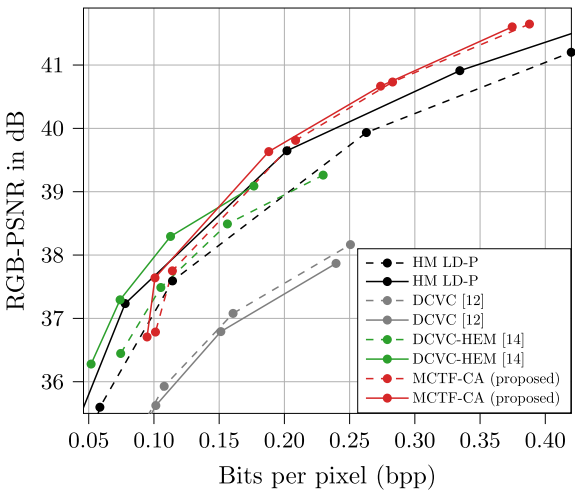

FIGURE 8. Rate-distortion evaluation on the MCL-JCV data set. Solid lines correspond to a GOP size of 8 and dashed lines to a GOP size of 4.

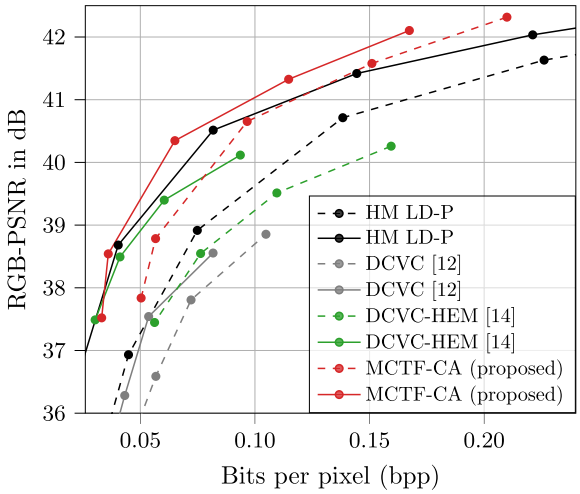

FIGURE 9. Rate-distortion evaluation on the HEVC E data set. Solid lines correspond to a GOP size of 8 and dashed lines to a GOP size of 4.

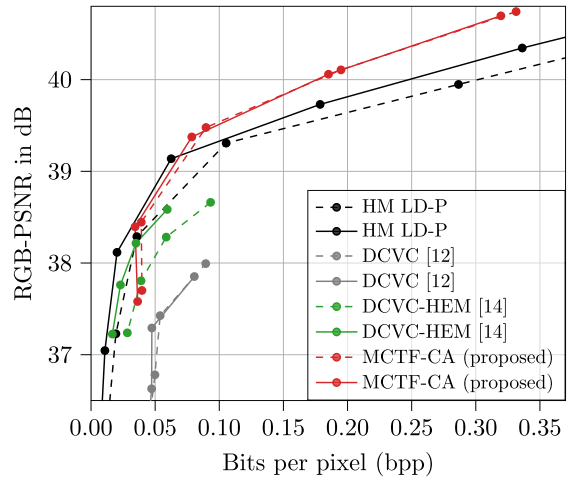

FIGURE 10. Rate-distortion evaluation on 3 sequences (*CityAlley, FlowerFocus, FlowerKids*) from the UVG 4K data set. Solid lines correspond to a GOP size of 8 and dashed lines to a GOP size of 4.

DCVC [12]. Figs. 7-10 show the rate-distortion curves for the UVG, MCL-JCV, HEVC E, and UVG 4K data sets, respectively. The dashed lines correspond to a GOP size of 4, whereas solid lines indicate a GOP size of 8.

Clearly, the conditional coder DCVC (gray) is not competitive with the remaining video coders. The approach performs better for a smaller GOP size of 4 compared to a GOP size of 8 on two data sets, which implies an error propagation issue. Its successor, DCVC-HEM (green), on the other hand, can effectively exploit temporal redundancies for the larger GOP size of 8. DCVC-HEM outperforms HM in lower bitrate ranges on UVG and MCL-JCV, whereas HM always performs better at higher rates.

The rate-distortion performance of the best-performing model, MCTF-CA (red), behaves in the opposite way: the higher the rate, the better the approach performs relative to HM. At higher rates, the model clearly outperforms HM for all data sets and GOP sizes. The performance degrades only for the MCTF-CA model at the lowest rate point ($\lambda = 0.007$) compared with the other rate points. The MCTF-CA model performs particularly well at high rates, owing to its invertible wavelet transforms. The perfect reconstruction property allows lossless compression without quantization, and therefore provides the capacity for high coding efficiency at high quality.

b: BJØNTEGAARD DELTA RATE

For a quantitative evaluation of the rate-distortion performance, the Bjøntegaard Delta (BD) rate savings of the learned video coders are measured using HM LD-P as an anchor. Note that the BD values need to be handled with caution because the available supporting points of DCVC-HEM and DCVC cover a limited bitrate and quality range. Thus, comparisons in terms of the BD metric can

TABLE 2. Rate-distortion evaluation on the UVG, MCL-JCV, HEVC E, and UVG 4K data sets for different GOP sizes. Average BD rate savings are provided relative to HM in LD-P configuration as an anchor.

(a) **GOP 4**

	DCVC	DCVC-HEM	MCTF-CA
UVG	+64.10%	-0.75%	**-21.48%**
MCL-JCV	+68.27%	-1.42%	**-12.63%**
HEVC E	+35.83%	+17.84%	**-26.20%**
UVG 4K	+207.75%	+56.53%	**-21.48%**

(b) **GOP 8**

	DCVC	DCVC-HEM	MCTF-CA
UVG	+134.50%	-3.93%	**-9.17%**
MCL-JCV	+131.30%	**-4.36** %	-0.41%
HEVC E	+96.68%	+13.31%	**-11.22%**
UVG 4K	+405.48%	+52.31%	**+3.70%**

be less reliable [46], and rate-distortion curves should be considered to obtain a complete picture. Therefore, using HM as an anchor avoids comparing the two conditional coders with the proposed method directly, but still perform comparisons over different bitrate and quality ranges. To cover the entire bitrate-distortion range of the learned video coders, HM is evaluated with Quantization Parameters (QP) values QP $= \{32, 27, 22, 19, 17, 15, 13\}$. The integration area for BD rate calculation is determined by the respective learned video coder, that is, by the minimum and maximum RGB-PSNR values obtained with the learned coder. Compared with the entire rate-distortion curve of HM, the overlap of the rate-distortion curve of DCVC-HEM with respect to the bitrate lies in the range of 14-29%. The overlap in terms of RGB-PSNR is between 42 and 46% depending on the data set and GOP size. Comparing the overlap of the rate-distortion curves of HM and MCTF-CA, the rate overlap is between 36 and 66%, whereas the distortion overlap ranges from 70 to 94 %. Hence, the MCTF models cover a larger rate-distortion range, as shown in Figs. 7-10.

Table 2 contains the BD measurements for all four data sets and for both GOP sizes. Over the entire bitrate range, DCVC-HEM performs best on the MCL-JCV data set for a GOP size of 8, achieving a BD rate reduction of approximately -4% compared to HM. In the remaining cases, MCTF-CA performs the best. It achieves BD rate savings of up to -21% and -9% on the UVG data set for GOP sizes of 4 and 8, respectively. On MCL-JCV, BD rate savings of -12% are obtained for a GOP size of 4. Furthermore, MCTF-CA achieves coding gains of -26% and -11% for GOP sizes of 4 and 8, respectively, on HEVC E. Overall, the high-resolution sequences from the UVG 4K data set are the most challenging for all learned video coders. MCTF-CA only achieves coding gains over HM for a GOP size of 4, but nevertheless performs favorably in comparison to the remaining learned coders.

TABLE 3. BD rate savings for each of the 7 UVG sequences over HM in LD-P configuration.

	GOP 4		GOP 8	
	DCVC-HEM	MCTF-CA	DCVC-HEM	MCTF-CA
HoneyBee	+16.72%	-52.38%	+17.60%	-50.29%
Bosphorus	-9.27%	-14.71%	-16.27%	-9.87%
Beauty	+337.29%	-64.37%	+373.47%	-63.63%
YachtRide	-18.21%	-5.69%	-22.40%	+1.48%
ShakeNDry	-3.52%	-29.06%	-0.46%	-21.44%
Jockey	+12.78%	+18.81%	+10.16%	+93.19%
ReadySteady	-12.17%	-10.93%	-19.35%	+14.16%

c: PER-SEQUENCE EVALUATION ON THE UVG DATA SET

The coding performance of DCVC-HEM and MCTF-CA is assessed for each of the seven sequences in the UVG data set. Table 3 provides BD rate savings relative to HM as an anchor. Independent of the GOP size, DCVC-HEM performs better than the proposed approach compared to HM for sequences with stronger motion, namely, *Jockey*, *ReadySteady*, and *YachtRide*. These sequences mostly contain relatively large translational motion. In contrast, the MCTF-CA approach performs best for sequences with high spatial detail and more irregular motion. For example, the approach achieves BD rate savings of over -63% for the *Beauty* scene, which is challenging because of moving hair. Here, DCVC-HEM struggles and is the least efficient compared to HM.

Overall, the per-sequence evaluation shows that MCTF leads to superior coding performance compared to an "IPPP..." coding order for specific scene contents. The following example of the *ShakeNDry* sequence illustrates the benefits of the temporal update operation. The scene has a static background, but contains challenging motion with flying water drops. With the MCTF-CA model, the first GOP of the sequence is coded with a GOP size of 8, that is, three temporal decomposition levels. The temporal updates help improve the coding efficiency of the temporal highpass frames at higher temporal decomposition levels: the highpass frames in the first, second, and third level require 0.38 bpp, 0.27 bpp, and 0.20 bpp at approximately 42.2 dB. As shown in Fig. 11(d)-(f), the highpass $h_{3,4}$ from temporal decomposition level three contains fewer prediction errors compared to the other levels, which leads to better coding efficiency. The application of two temporal update operations (see Fig. 11 (c)) creates a better representation for the prediction compared to the original frame in Fig. 11 (b) through lowpass filtering along the motion trajectory.

When comparing the rate-distortion curves of MCTF-CA for every sequence of the UVG data set (cf. Fig. 12), the *ShakeNDry* sequence is one of the most challenging sequences next to the *Beauty* sequence. Fig. 12 provides the motion-compensated prediction quality in terms of PSNR and maximum motion vector length in pixels averaged over all 96 evaluated frames for each sequence. These values are computed using a SPyNet model trained on Vimeo90K without considering motion vector compression. These measurements show that a high prediction quality of over 48 dB and

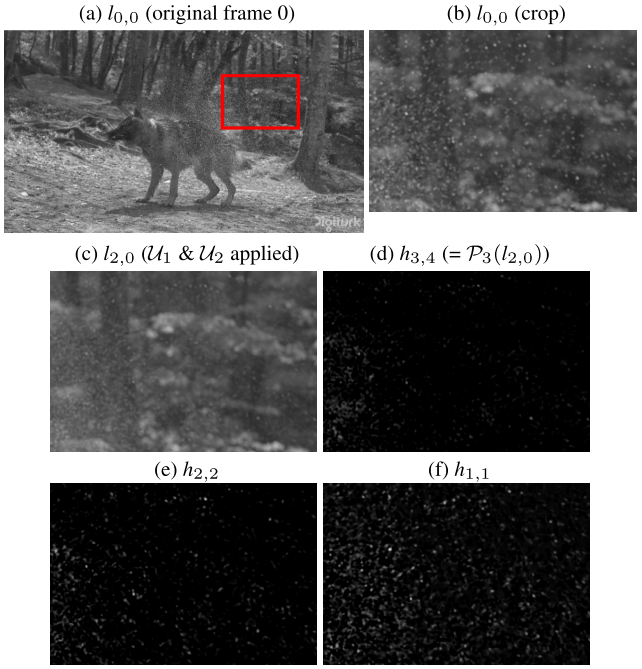

FIGURE 11. Impact of the temporal update operation. Subfig. (a) shows the first frame of the *ShakeNDry* sequence from the UVG data set. (d)-(f) depict temporal highpass frames coded in different temporal decomposition levels by a MCTF-CA model ($\lambda = 0.08$, GOP size 8). The highpass frame $h_{3,4}$ from the highest temporal decomposition level has less prediction errors compared to $h_{2,2}$ and $h_{1,1}$ (black corresponds to zeros). This is because $h_{3,4}$ is predicted from $l_{2,0}$ shown in (c). Here, the application of temporal updates to $l_{2,0}$ improves the prediction and thus coding efficiency.

TABLE 4. Complexity comparison of learned video coders for an input size of 1920 × 1080 in terms of model size and kilo multiply-accumulate operations per pixel (kMAC/px).

	DCVC	DCVC-HEM	MCTF-CA
Model size	32 MB	70 MB	90 MB
kMAC/px	1167	1673	3554

relatively small motion (*HoneyBee*, *Bosphorus*) are associated with the best rate-distortion performance of MCTF-CA. However, a lower prediction quality and larger motion do not necessarily lead to poor rate-distortion performance; for example, MCTF-CA performs better on the *Jockey* sequence than on the *Beauty* sequence because factors such as high spatial detail contained in a sequence influence the coding efficiency as well.

d: COMPLEXITY

The computational complexity of the MCTF-based approach is assessed in terms of model size and kilo multiply-accumulate operations per pixel (kMAC/px). As shown in Table 4, the MCTF-CA approach is more complex with respect to both model size and kMACs/px. Note that most of the model complexity of MCTF-CA is attributed to the temporal subband coder iWave++. For a GOP size of 8, the MCTF modules only account for 29 % of the model size and 12 % of the required kMACs/px. Because of the dedicated MCTF stages for every temporal decomposition level, the MCTF modules have a larger influence on the model size relative to MACs.

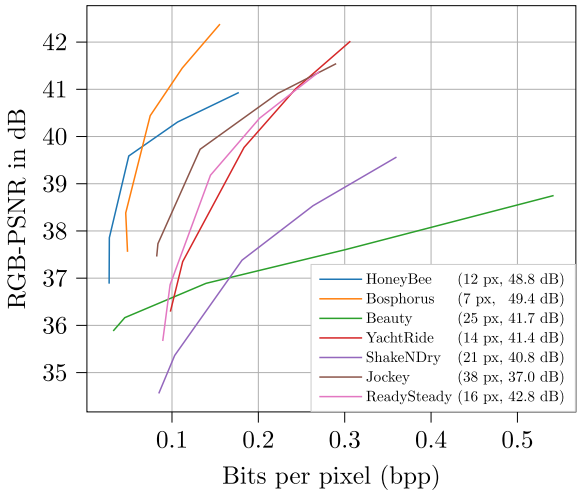

FIGURE 12. Comparison of the rate-distortion curves of MCTF-CA (GOP size of 8) for every sequence of the UVG data set. For each sequence, the motion strength in pixels (px) and motion-compensated prediction quality in dB averaged over all frames are provided. For these measurements, the motion vectors between successive frames required for motion compensation are estimated using a SPyNet model. Thereby, the motion strength for a single frame is measured as the maximum motion vector length in horizontal or vertical direction.

TABLE 5. Rate-distortion evaluation on the UVG and MCL-JCV data sets for different GOP sizes. Average BD rate savings are provided relative to the baseline MCTF model as an anchor.

	GOP 4		GOP 8	
	UVG	MCL-JCV	UVG	MCL-JCV
MCTF-Single	+17.26%	+16.65%	+34.51%	+29.60%
MCTF-DS	-2.90%	-2.11%	-3.46%	+1.05 %
MCTF-CA	**-4.68%**	**-8.74%**	**-10.15%**	**-14.94%**

2) ABLATION STUDY: MCTF CONFIGURATION

In the following section, several MCTF coder configurations are examined. In doing so, the benefits of the proposed downsampling strategy and content-adaptive MCTF approach are evaluated.

TABLE 6. BD rate savings for each of the 7 UVG sequences over the baseline MCTF model as an anchor.

	GOP 4		GOP 8	
	MCTF-DS	MCTF-CA	MCTF-DS	MCTF-CA
HoneyBee	+0.25%	-0.33%	+9.83%	-0.01%
Bosphorus	-1.76%	-2.36%	-0.02%	-0.02%
Beauty	-0.93%	-1.99%	+3.45%	-8.31%
YachtRide	-4.22%	-4.22%	-5.01%	-12.17%
ShakeNDry	-0.81%	-0.81%	+0.34%	-0.13%
Jockey	-7.17%	-16.44%	-7.36%	-25.15%
ReadySteady	-3.86%	-4.39%	-12.39%	-12.64%

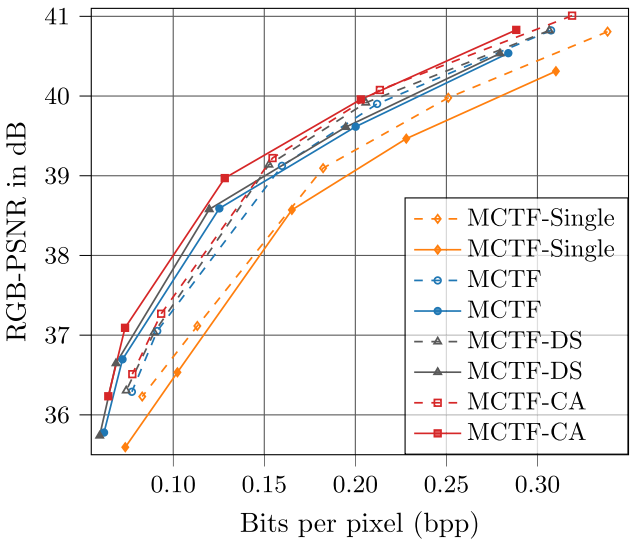

FIGURE 13. Rate-distortion evaluation on the UVG data set. Solid lines correspond to a GOP size of 8 and dashed lines to a GOP size of 4. ***MCTF-Single***: Same MCTF stage for all temporal decomposition levels. ***MCTF***: Different MCTF stages for each level. ***MCTF-DS***: Different MCTF stages with downsampling strategy during inference. ***MCTF-CA***: Content adaptive MCTF. ***Best to be viewed enlarged on a screen.***

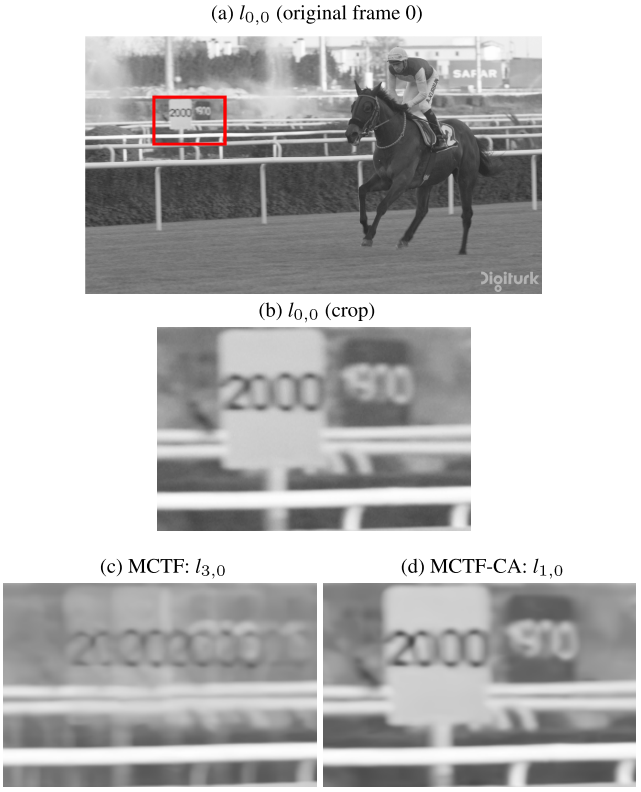

FIGURE 14. Content adaptive MCTF prevents ghosting. Subfig. (a) shows the first frame of a GOP of size 8 from the *Jockey* sequence. The MCTF model in (c) codes the first frame as $l_{3,0}$ in third temporal decomposition level, which leads to ghosting due to the large motion in the scene ($C_{8,\text{GOP8}} = 1.74$). MCTF-CA in (d) mitigates ghosting by choosing a GOP size of 2 and transmitting the first frame as $l_{1,0}$ in the first temporal decomposition level ($C_{8,\text{GOP2}} = 1.57$).

a: MULTIPLE MCTF STAGES

First, a single MCTF stage (''MCTF-Single'') is evaluated and compared with multiple MCTF stages. The latter uses dedicated MCTF modules for each temporal decomposition level, that is, different DN, motion estimation, and motion vector compression networks for every level. Table 5 compares the MCTF-Single model with the MCTF model with multiple MCTF stages as an anchor. The MCTF-Single model is obtained at the end of training stage three (cf. Table 1). It is included in the evaluation, because it corresponds to the standard approach commonly used in traditional MCTF.

On both data sets, MCTF-Single results in a BD rate degradation of over +16% and +29% for GOP sizes of 4 and 8, respectively. Therefore, multiple MCTF stages are necessary to achieve improved rate-distortion performance for higher temporal decomposition levels with larger frame distances.

The impact of multiple MCTF stages on the rate-distortion curves for the UVG data set is illustrated in Fig. 13. The models with multiple MCTF stages (blue) clearly outperform a single stage (orange), independent of the GOP size.

b: DOWNSAMPLING STRATEGY (MCTF-DS)

Next, the MCTF-DS approach introduced in Section IV-C is evaluated. On average, the MCTF-DS models (gray) lead to a reduced bitrate at approximately the same quality as the baseline models (blue), as shown in Fig. 13. The bitrate savings are due to the smaller spatial resolution of the motion vectors, which requires a lower rate. At the same time, there is no significant quality degradation, and for some rate points, the quality is even slightly improved. On average, MCTF-DS leads to coding gains between 2 and 3%, measured in terms of BD rate, compared to the MCTF model with multiple stages as an anchor (cf. Table 5). However, MCTF-DS degrades the performance on the MCL-JCV data set for a GOP size of 8.

Table 6 provides the BD rate evaluation for each sequence of the UVG data set. For scenes containing larger motion (*ReadySteady*, *YachtRide*, *Jockey*), MCTF-DS achieves BD rate savings of up to −4% and −12% for a GOP size of 4 and 8, respectively, compared to the MCTF model with multiple MCTF stages. Consequently, MCTF-DS improves the performance for larger motion. For the *HoneyBee* sequence with a small moving object and high spatial detail, the downsampling strategy leads to BD rate increases of 0.5% and 10% for GOP sizes of 4 and 8, respectively. This shows that although the downsampling strategy leads to improved performance for most sequences, a content-adaptive mechanism is required.

c: CONTENT-ADAPTIVE MCTF (MCTF-CA)

The MTCF-CA approach explained in Section IV-D overcomes the disadvantages of MCTF-DS for some motion types and scene contents. As can be seen in Table 5, MCTF-CA performs best on all data sets and GOP sizes. In particular, for a GOP size of 8, MCTF-CA provides average BD rate savings of at least 10% compared to the MCTF model with multiple MCTF stages as an anchor.

A detailed evaluation on every sequence of the UVG data set provided in Table 6 shows that for a GOP size of 4, MCTF-CA improves over MCTF-DS for 5 out of 7 sequences. For the remaining two sequences, MCTF-DS is already optimal. However, for a GOP size of 8 more options for MCTF-CA are available and MCTF-DS is only optimal for the *Bosphorus* sequence, which contains relatively easy translational motion. For the remaining sequences, a content-adaptive approach leads to considerable improvements in terms of BD rate; for example, MCTF-CA achieves BD rate savings of -12% and -25% on the *YachtRide* and *Jockey* sequences, respectively. Furthermore, MCTF-CA prevents the use of the downsampling strategy for sequences where it degrades rate-distortion performance, for example, for the *HoneyBee* and *Beauty* sequences containing high spatial detail. Therefore, content-adaptive temporal scaling is clearly advantageous in terms of rate-distortion performance, because the motion types are highly dependent on the scene content.

Fig. 14 provides an example of the benefit of MCTF-CA: the *Jockey* sequence from the UVG data set contains strong motion, which leads to ghosting for some GOPs (cf. Fig. 14(c)) when processing the sequence with a uniform temporal decomposition, that is, a constant GOP size of 8 with the MCTF model. MCTF-CA adaptively chooses a smaller GOP size if ghosting harms the coding costs. As can be seen in Fig. 14(d), MCTF-CA prevents ghosting by determining a GOP size of 2, which can be coded most efficiently.

VI. CONCLUSION

This paper introduced the first end-to-end trainable wavelet video coder based on MCTF. It presented a training strategy that considers multiple temporal decomposition levels during training. Moreover, a downsampling strategy was proposed as a first solution for handling larger temporal displacements in MCTF. The novel content-adaptive MCTF enables the proposed method to adapt to different motion types in each sequence. The experimental results show that the learned MCTF video coder exhibits promising rate-distortion performance, especially for higher bitrates. On the UVG data set, the MCTF-CA method achieves average BD rate savings of -21% and -9% for GOP sizes of 4 and 8, respectively, compared to HM. Thereby, it clearly outperforms the state-of-the-art video coder DCVC-HEM [14].

There are various possibilities for improvement as an initial version of a learned wavelet video coder. First, one could examine a different temporal subband coder required for practical usage because the autoregressive context model of iWave++ prohibits parallelization. Second, the MCTF structure requires extensions to handle more diverse motion types and GOP sizes of 16 and higher. Because the maximum frame distance doubles with every additional temporal decomposition level, motion estimation is considerably more challenging for, for example, a GOP size of 16 with a frame distance of 8. Therefore, bidirectional motion estimation and methods for overcoming the limitations of short-sequence training sets for larger GOP-size compression could be investigated. To mitigate ghosting for larger GOP sizes, an adaptive choice of a truncated DWT without temporal update [47] could be beneficial. Furthermore, the complexity of content-adaptive MCTF can be limited by using a predictor for choosing the adaptive MCTF option.

The MCTF-based approach provides an explainable and scalable alternative to common autoencoder-based video coders. This paper made the first steps to enable further development of this important direction of research.

REFERENCES

[1] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10998–11007.
[2] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3292–3308, Oct. 2021.
[3] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, "Learning for video compression with recurrent auto-encoder and recurrent probability model," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 388–401, Feb. 2021.
[4] Z. Hu, G. Lu, and D. Xu, "FVC: A new framework towards deep video compression in feature space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1502–1511.
[5] Z. Hu, G. Lu, J. Guo, S. Liu, W. Jiang, and D. Xu, "Coarse-to-fine deep video coding with hyperprior-guided mode prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5921–5930.
[6] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8503–8512.
[7] O. Rippel, A. G. Anderson, K. Tatwawadi, S. Nair, C. Lytle, and L. Bourdev, "ELF-VC: Efficient learned flexible-rate video coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14479–14488.
[8] W. Park and M. Kim, "Deep predictive video compression using mode-selective uni- and bi-directional predictions based on multi-frame hypothesis," *IEEE Access*, vol. 9, pp. 72–85, 2021.
[9] R. Yang, R. Timofte, and L. Van Gool, "Advancing learned video compression with in-loop frame prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2410–2423, May 2023.
[10] N. Van Thang and L. Van Bang, "Hierarchical random access coding for deep neural video compression," *IEEE Access*, vol. 11, pp. 57494–57502, 2023.
[11] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.
[12] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, Dec. 2021, pp. 18114–18125.
[13] X. Sheng, J. Li, B. Li, L. Li, D. Liu, and Y. Lu, "Temporal context mining for learned video compression," *IEEE Trans. Multimedia*, early access, Nov. 8, 2022, doi: 10.1109/TMM.2022.3220421.
[14] J. Li, B. Li, and Y. Lu, "Hybrid spatial–temporal entropy modelling for neural video compression," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1503–1511.
[15] Y.-H. Ho, C.-P. Chang, P.-Y. Chen, A. Gnutti, and W.-H. Peng, "CANF-VC: Conditional augmented normalizing flows for video compression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2022, pp. 207–223.
[16] F. Mentzer et al., "VCT: A video compression transformer," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 35, Nov. 2022, pp. 13091–13103.
[17] J. Xiang, K. Tian, and J. Zhang, "MIMT: Masked image modeling transformer for video compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2023, pp. 1–17.
[18] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 559–571, Sep. 1994.
[19] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[20] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[21] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017, pp. 1–27.
[22] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2018, pp. 1–47.
[23] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Dec. 2018, pp. 10771–10780.
[24] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear transform coding," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 339–353, Feb. 2021.
[25] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-end optimized versatile image compression with wavelet-like transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1247–1263, Mar. 2022.
[26] F. Brand, J. Seiler, and A. Kaup, "On benefits and challenges of conditional interframe video coding in light of information theory," in *Proc. Picture Coding Symp. (PCS)*, Dec. 2022, pp. 289–293.
[27] F. Brand, J. Seiler, and A. Kaup, "Intra-frame coding using a conditional autoencoder," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 354–365, Feb. 2021.
[28] T. Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "ModeNet: Mode selection network for learned video coding," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2020, pp. 1–6.
[29] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. New York, NY, USA: Springer, 2002.
[30] T. Borer, "WHP 238—The VC-2 low delay video codec," Brit. Broadcast. Corp. (BBC), London, U.K., Tech. Rep. WHP 238, Aug. 2013.
[31] W. Sweldens, "Lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Proc. SPIE*, vol. 2569, pp. 68–79, Sep. 1995.
[32] H. Ma, D. Liu, R. Xiong, and F. Wu, "IWave: CNN-based wavelet-like transform for image compression," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1667–1679, Jul. 2020.
[33] D. Xue, H. Ma, L. Li, D. Liu, and Z. Xiong, "IWave3D: End-to-end brain image compression with trainable 3-D wavelet transform," in *Proc. Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2021, pp. 1–5.
[34] D. Xue, H. Ma, L. Li, D. Liu, and Z. Xiong, "AiWave: Volumetric image compression with 3-D trained affine wavelet-like transform," *IEEE Trans. Med. Imag.*, vol. 42, no. 3, pp. 606–618, Mar. 2023.
[35] C. Dong, H. Ma, D. Liu, and J. W. Woods, "Wavelet-based learned scalable video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2022, pp. 3190–3194.
[36] Y. Liu and J. W. Woods, "New and efficient interframe extensions of EZBC and JPEG 2000," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.
[37] A. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 1997, pp. 596–599.
[38] D. Lanz, F. Schilling, and A. Kaup, "Scalable lossless coding of dynamic medical CT data using motion compensated wavelet lifting with denoised prediction and update," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.
[39] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2720–2729.
[40] D. Lanz, C. Herbert, and A. Kaup, "Content adaptive wavelet lifting for scalable lossless video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1782–1786.
[41] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Feb. 2019.
[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Nov. 2019, pp. 1–8.
[43] A. Meyer and A. Kaup, "A novel cross-component context model for end-to-end wavelet image coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
[44] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 297–302.
[45] H. Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo, "MCL-JCV: A JND-based H.264/AVC video quality assessment dataset," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1509–1513.
[46] C. Herglotz, H. Och, A. Meyer, G. Ramasubbu, L. Eichermüller, M. Kränzler, F. Brand, K. Fischer, D. T. Nguyen, A. Regensky, and A. Kaup, "The Bjøntegaard bible—Why your way of comparing video codecs may be wrong," Apr. 2023, *arXiv:2304.12852*.
[47] D. S. Turaga, M. van der Schaar, Y. Andreopoulos, A. Munteanu, and P. Schelkens, "Unconstrained motion compensated temporal filtering (UMCTF) for efficient and flexible interframe wavelet video coding," *Signal Process., Image Commun.*, vol. 20, no. 1, pp. 1–19, Jan. 2005.

ANNA MEYER (Graduate Student Member, IEEE) received the master's degree in advanced signal processing and communications engineering from Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany, in 2021.

During her master's, she worked on multispectral image compression and few-shot object detection for artworks. Since 2021, she has been a Researcher with the Chair of Multimedia Communications and Signal Processing, FAU, where she conducts research on wavelet video compression and deep learning.

FABIAN BRAND (Graduate Student Member, IEEE) received the master's degree in electrical engineering from Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany, in 2018.

During his bachelor's, he worked on methods for frame-rate-conversion of video sequences, and during his master's, he researched automated harmonic analysis of classical music and style classification. Since 2019, he has been a Researcher with the Chair of Multimedia Communications and Signal Processing, FAU, where he conducts research on methods for video compression and deep learning. For his work, among others, he received the Best Paper Award of the Picture Coding Symposium (PCS) 2019.

ANDRÉ KAUP (Fellow, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from RWTH Aachen University, Aachen, Germany, in 1989 and 1995, respectively.

He joined Siemens Corporate Technology, Munich, Germany, in 1995, and became the Head of the Mobile Applications and Services Group, in 1999. Since 2001, he has been a Full Professor and the Head of the Chair of Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany. From 2005 to 2007, he was a Vice Speaker of the DFG Collaborative Research Center 603. From 2015 to 2017, he was the Head of the Department of Electrical Engineering and the Vice Dean of the Faculty of Engineering, FAU. He has authored around 450 journal and conference papers and has over 120 patents granted or pending. His research interests include image and video signal processing and coding and multimedia communication.

Dr. Kaup is a member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee, the Scientific Advisory Board of the German VDE/ITG, and the Bavarian Academy of Sciences. He is a member of the Editorial Board of the IEEE Circuits and Systems Magazine. He was a Siemens Inventor of the Year 1998 and received the 1999 ITG Award. He received several IEEE best paper awards, including the Paul Dan Cristea Special Award, in 2013, and his group won the Grand Video Compression Challenge from the Picture Coding Symposium, in 2013. The Faculty of Engineering with FAU and the State of Bavaria honored him with teaching awards, in 2015 and 2020, respectively. He served as an Associate Editor for IEEE Transactions on Circuits and Systems for Video Technology. He was the Guest Editor of the IEEE Journal of Selected Topics in Signal Processing.