Region-Based Template Matching Prediction for Intra Coding

Gayathri Venugopal, Karsten Müller, *Senior Member, IEEE*, Jonathan Pfaff, Heiko Schwarz, Detlev Marpe, *Fellow, IEEE*, and Thomas Wiegand, *Fellow, IEEE*

***Abstract*— Copy prediction is a renowned category of prediction techniques in video coding where the current block is predicted by copying the samples from a similar block that is present somewhere in the already decoded stream of samples. Motion-compensated prediction, intra block copy, template matching prediction etc. are examples. While the displacement information of the similar block is transmitted to the decoder in the bit-stream in the first two approaches, it is derived at the decoder in the last one by repeating the same search algorithm which was carried out at the encoder. Region-based template matching is a recently developed prediction algorithm that is an advanced form of standard template matching. In this method, the reference area is partitioned into multiple regions and the region to be searched for the similar block(s) is conveyed to the decoder in the bit-stream. Further, its final prediction signal is a linear combination of already decoded similar blocks from the given region. It was demonstrated in previous publications that region-based template matching is capable of achieving coding efficiency improvements for intra as well as inter-picture coding with considerably less decoder complexity than conventional template matching. In this paper, a theoretical justification for region-based template matching prediction subject to experimental data is presented. Additionally, the test results of the aforementioned method on the latest H.266/Versatile Video Coding (VVC) test model (version VTM-14.0) yield an average Bjøntegaard-Delta (BD) bit-rate savings of −0.75% using all intra (AI) configuration with 130% encoder run-time and 104% decoder run-time for a particular parameter selection.**

***Index Terms*— Template matching, video coding, intra-picture prediction, AVC, HEVC, VVC.**

I. Introduction

In A block-based hybrid video coding standard like H.264/Advanced Video Coding (AVC), H.265/High Efficiency Video Coding (HEVC) or the recent H.266/Versatile Video Coding (VVC), the pictures of an input video sequence are partitioned into blocks. Then predictive coding (intra or inter) is applied together with transform coding and quantization, followed by the entropy coding of the prediction data and quantization indexes [1], [2], [3], [4]. While intra-picture prediction is used to take advantage of the spatial redundancy in the pictures for video compression, inter-picture prediction is employed for exploiting the temporal redundancy between the pictures. Typically, the former is achieved by extrapolating the boundary samples of the current block[1] in a predefined manner. The DC, PLANAR, and ANGULAR modes in H.264/AVC and its descendants are instances of this type of prediction. On the other hand, inter-picture prediction traditionally relies on motion-compensated prediction (MCP) which is a popular example of copy prediction methods.

A copy prediction approach assumes that a similar block (termed as predictor block) to the current block is present in the already decoded stream of samples. This block is found by initiating a search algorithm in the reference area, and later, its samples are copied to the current block as the prediction signal. MCP, intra block copy (IBC), template matching prediction (TMP) etc. are examples of this type of prediction. In MCP, the predictor block is identified at the encoder-side through a block-matching (BM)-based search mechanism where the original block is compared against the potential reference blocks in the reference area. Error metrics like the sum of absolute differences (SAD) or sum of squared differences (SSD) are typically used for measuring the similarity between the blocks, and the reference block that leads to the least distortion is treated as the predictor block. In a more efficient encoder implementation, the bits for transmitting the motion data (i.e., the motion vector (MV) and index to the reference picture) related to the reference block are also taken into account and the selection that results in the least rate-distortion (RD) cost is chosen as the predictor block. At last, the motion information of the predictor block is transmitted to the decoder in the bit-stream for the final reconstruction of the current block. The second example, IBC, is an intra-picture prediction method that is analogous to MCP with the difference that the reference area is the current partially reconstructed picture [3], [4], [5], [6], [7], [8], [9], [10], [11]. In the case of TMP, the motion information of the predictor block is not transmitted explicitly to the decoder as in MCP and IBC, instead, it is obtained by initiating an identical template matching (TM) search at the encoder and decoder [12]. In detail, a match for the template (conventionally, the neighbouring samples to the top, left and top-left corner of a block is regarded as its *template*) of the current block is found from the reference area by minimizing an error metric (like SAD, SSD etc.), and then the block corresponding

Manuscript received 21 December 2021; revised 19 September 2022 and 11 November 2022; accepted 16 December 2022. Date of publication 6 January 2023; date of current version 13 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sergio De Faria. *(Corresponding author: Gayathri Venugopal.)*

Gayathri Venugopal, Karsten Müller, Jonathan Pfaff, and Detlev Marpe are with the Fraunhofer HHI, 10587 Berlin, Germany (e-mail: gayathri.venugopal@hhi.fraunhofer.de).

Heiko Schwarz is with the Fraunhofer HHI, 10587 Berlin, Germany, and also with the Free University of Berlin, 14195 Berlin, Germany.

Thomas Wiegand is with Fraunhofer Heinrich Hertz Institute (HHI), 10587 Berlin, Germany, and also with the Information Technology, Technical University of Berlin, 10623 Berlin, Germany.

Digital Object Identifier 10.1109/TIP.2022.3233184

[1]The block under consideration for compression/decompression is regarded as current block. Its uncompressed block is termed original block.

to the best template match is considered as the predictor block of the current block. Finally, the samples of the predictor block are copied to the current block as the prediction. Additionally, if multiple template matches are allowed, the average of the samples of the corresponding blocks is typically used as the final prediction signal. The major drawback of TMP is that the entire search process for the predictor block(s) has to be repeated at the decoder-side also, resulting in a large number of computations there.

TMP was proposed for video coding for the first time in [13]. It was for intra-picture prediction. Later, many research activities were followed for improving its coding efficiency [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. Further, its application to inter-picture prediction was also studied [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35]. These studies and publications on TMP were primarily focused on attaining further compression efficiency, while the high decoder complexity of TMP was mostly ignored.

Region-based template matching prediction (RTMP), which is the subject of this paper, is an advanced form of standard TMP. Previous publications demonstrated that RTMP has higher coding efficiency and lower decoder complexity than its predecessor [36], [37], [38], [39]. In RTMP, the reference area is partitioned into multiple regions and the final prediction signal is a linear combination of the predictor blocks obtained through TM search from the given region. At the encoder, an independent prediction signal from each region is obtained and the best among them is identified using the standard rate-distortion optimization[2] (RDO) or a similar algorithm. Then, the index of the region that gives the best prediction is conveyed to the decoder in the bit-stream. Later, at the decoder, the TM search routine is carried out for the predictors only in the region corresponding to the parsed index. Thus, the prediction efficiency and decoder computational efficiency of RTMP are improved when compared to TMP.

While previous publications [36], [37], [38], [39] targeted algorithm description and optimization of RTMP for practical applications, this paper focuses on the theoretical background of RTMP. Accordingly, a theoretical justification for the compression efficiency of RTMP in the context of intra coding (however, it holds for inter coding also) is detailed in the next section. After that, the realization of RTMP for a block-based hybrid video codec is explained in section III. It is then followed by the experimental results of RTMP against H.266/VVC in section IV. At last, the conclusions from this publication are given in section V. Note that here after in this paper, RTMP for intra coding is abbreviated as intra-RTM. Similarly, TMP for intra coding is shortened as intra-TM.

[2]In a standard encoder, RDO is utilized to identify the best coding parameters among the varied selection of coding tools like coding modes, prediction parameters, quantization parameters etc. for a given coding efficiency. Here, the rate and distortion resulting from a given method are determined. Finally, the coding parameters that minimize the RD cost over a set of choices are considered as the final selection and their side information is transmitted to the decoder [2].

II. Analysis of Region-Based Template Matching Prediction

This section is divided into two parts. Firstly, the prediction efficiency of RTMP is examined in subsection II-A. Later, the RD performance of RTMP is evaluated in subsection II-B. Note that it is assumed throughout this section that TMP, RTMP and IBC (or MCP, in case of inter coding) have the same reference area, and they use the same error metric in their corresponding search algorithms. Additionally, by distortion, we refer to prediction distortion, i.e., the mean squared error (MSE) between the original block and the prediction block generated by RTMP or TMP or any other method.

A. Prediction Efficiency of RTMP

In traditional TMP, the block corresponding to the best template match in the reference reconstructed area is regarded as the predictor block of the current block, and the distortion is measured using this block. Hence, it can be deduced that TMP is inadequate in obtaining the best (or most similar) predictor on all cases, unlike IBC which uses the original block to find its predictor block. This can be explained through statistical dependencies, as the current block is consistently correlated to its original block, however not necessarily to its template in the same way. Thus, the predictor from IBC is the optimal one among TMP and IBC. This leads to the further inference that the displacement error associated with TMP is always greater than that from IBC. Note that the displacement error from a copy prediction method is defined as the difference between the displacement of the predictor obtained through that method and the true displacement. Therefore, if ΔIBC and ΔTMP are the displacement errors from IBC and TMP respectively, then

$$\Delta\text{IBC} \leq \Delta\text{TMP} \tag{1}$$

where

$$\Delta\text{IBC} = m^t - m^{\text{IBC}} \tag{2}$$

and

$$\Delta\text{TMP} = m^t - m^{\text{TMP}}. \tag{3}$$

$m^{\text{IBC}} = (m_x^{\text{IBC}}, m_y^{\text{IBC}})$ and $m^{\text{TMP}} = (m_x^{\text{TMP}}, m_y^{\text{TMP}})$ are the block vectors (BVs) of the current block B_c from IBC and TMP respectively. $m^t = (m_x^t, m_y^t)$ is the true BV associated with the current block B_c. Note that a BV gives the displacement of the predictor block from the current block and it is analogous to MV of MCP. It is clear from the relationship in Eq. (1) that the prediction efficiency from TMP is less than the corresponding efficiency from IBC. Therefore, this subsection describes how the prediction efficiency of TMP can be improved through the transition to RTMP.

In RTMP, the given reference area (or search window) is partitioned into multiple regions such that an independent prediction can be obtained from each region. Further, the region with the best prediction is identified through RDO (or any other similar mechanism) at the encoder-side, and the index of the chosen region is transmitted to the decoder in the bit-stream. Now, at the decoder-side, the TM search process for the predictor is carried out only in the region corresponding

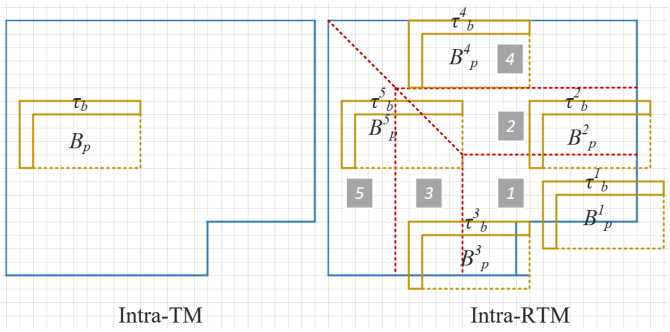


Fig. 1. Example illustration of intra-TM and intra-RTM (with the number of regions $n_r = 5$) predictions. The search window is marked by a solid blue line and the regions by dotted red lines. The predictor block and best template from intra-TM are represented by B_p and τ_b respectively. Similarly, the predictor block and best template from each region with index ν of intra-RTM are represented by B^ν_p and τ^ν_b respectively. The region index ν is highlighted by gray-encased numbers.

to the parsed index. Finally, the prediction signal is generated using the predictor from that region.

An example illustration of intra-TM and intra-RTM (with the number of regions $n_r = 5$) is given in Fig. 1. The predictor block and best template from intra-TM are represented by B_p and τ_b respectively. Similarly, the predictor block and best template from each region with index ν of intra-RTM are represented by B^ν_p and τ^ν_b respectively. The partitioning scheme for intra-RTM is based on a preliminary analysis of the position of intra-TM predictors against the current block. It was observed that most of the predictors originate from the immediate neighbourhood of the current block. Accordingly, in intra-RTM, the first region is treated as the *most probable region* and it is not partitioned. Moreover, a separate flag is assigned to this region in the region index coding (refer subsection III-E). The rest of the regions are partitioned diagonally. Besides, detailed investigatory tests on varied region shapes have indicated that using only vertical or horizontal regions degrades the performance of intra-RTM since such a partitioning scheme does not well accommodate the immediate neighbourhood of the current block with one *most probable region* in most occasions [40].

As shown in Fig. 1, only one prediction from the entire reference area is obtained in TMP. Further, no side information needs to be transmitted to the decoder, however the predictor estimation in the entire reference area needs to be repeated there. On the other hand, in RTMP, the region that gives the best prediction in the reference area is identified at the encoder and its index is transmitted to the decoder. In general, RTMP operates as a two-phase method at the encoder-side; In the first phase, the predictor estimation process in each region is carried out one-by-one using the TM search routine. In the second phase, the prediction block from each region is generated using the estimated predictor and it is evaluated through the standard RDO or a similar approach. Finally, the region that gives the least distortion (for RDO, the least RD cost) is considered as the best region and its index is transmitted to the decoder for the final reconstruction of the block.

Additionally, the second phase is comparable to a BM-based displacement (or motion) estimation stage as in IBC (or MCP), however, with a fewer number of reference blocks that were obtained from the first phase. Regarding the encoder complexity, the standard RDO approach in the second phase of RTMP would increase its computations considerably due to the inclusion of the quantization and transform coding steps. However, it is not mandatory to use the standard RDO algorithm here. A simplified RD cost estimation (for example, using Hadamard transform) or a distortion-only cost estimation can give a better gain-complexity trade-off [38], [39].

The distortion from RTMP is always smaller than that from TMP. First, RTMP can also always find the global best template match as in TMP and thus the distortion from RTMP is not greater than the one from TMP. On the other hand, assume for simplicity that RTMP is operated with two regions Γ_1 and Γ_2 and thus, for the comparison, TMP operates with the single region $\Gamma_1 \sqcup \Gamma_2$. Then it might very well happen that the global best template match is located in region Γ_1, while the distortion is smaller for TM search restricted to Γ_2 than for TM search restricted to Γ_1. Thus, in this case, the distortion from RTMP would be smaller than that from TMP. I.e., by testing the predictors related to the region-wise local best template matches in RTMP, the process of BM is mimicked and a predictor closer to or the same as the global best block match is achieved. Hence, the displacement error from RTMP is smaller than the corresponding error from TMP. The following experimental analysis provides evidence for this. First of all, the prediction methods of IBC, TMP, and RTMP (for different numbers of regions n_r) are implemented at the encoder of the VVC test model (version VTM-2.0.1) [41]. Next, the corresponding predictor blocks from each of the approaches are obtained for the same reference area. Then, the relative displacement of the predictor of TMP against IBC (assuming IBC has the true displacement) is calculated as,

$$\begin{aligned} \Delta\mathrm{TMP}_x &= \mathrm{IBC}_x - \mathrm{TMP}_x \\ \Delta\mathrm{TMP}_y &= \mathrm{IBC}_y - \mathrm{TMP}_y \end{aligned} \tag{4}$$

where the predictor from TMP and IBC is assumed to be located at $(\mathrm{TMP}_x, \mathrm{TMP}_y)$ and $(\mathrm{IBC}_x, \mathrm{IBC}_y)$ respectively. Similarly, if the predictor from RTMP is located at $(\mathrm{RTMP}_x, \mathrm{RTMP}_y)$, then the relative displacement of the predictor of RTMP against IBC is as given below.

$$\begin{aligned} \Delta\mathrm{RTMP}_x &= \mathrm{IBC}_x - \mathrm{RTMP}_x \\ \Delta\mathrm{RTMP}_y &= \mathrm{IBC}_y - \mathrm{RTMP}_y \end{aligned} \tag{5}$$

The number of occurrences of TMP predictor blocks with respect to the IBC predictor blocks inside a 70×70 window is measured using Eq. (4) for 1 frame (using all intra (AI) configuration of the common test conditions (CTC) defined for standard dynamic range (SDR) content by the Joint Video Experts Team (JVET) [42]) with quantization parameter (QP) equal to 22. For simplicity, all experiments in this section are restricted to 4×4 blocks. Then, the percentage of occurrences at $\Delta\mathrm{TMP}_x = 0, \Delta\mathrm{TMP}_y = 0$ is obtained. This indicates how often the predictor from TMP is the same as the predictor from IBC. Next, the number of occurrences of RTMP predictor blocks with that to IBC's is measured for $n_r \in \{3, 5, 9, 17\}$ using Eq. (5), and the corresponding percentage of occurrences at $\Delta\mathrm{RTMP}_x = 0, \Delta\mathrm{RTMP}_y = 0$ is gathered. The related average distortion (the average MSE) in each case is also

obtained. At last, the average distortion related to IBC is also measured which is for $n_r = n_s$ where n_s is the number of samples in the reference area. In the experiment, $n_s = 1140$ as $n_s = ((\zeta + M) \times (\zeta + N) - (M \times N))$ for the search window as shown in Fig. 1 and $\zeta = 30$, $M = 4$, $N = 4$. (Refer section III for more explanations on ζ.) When $n_r = n_s$, every sample in the reference area is a region, and the block corresponding to the sample is the predictor. In that instance, the TM search in the first phase of RTMP does not apply and only the BM search in the second phase is carried out. Thus, RTMP becomes equivalent to IBC, assuming the error metric in IBC and second phase of RTMP are the same. Moreover, if RDO is used in the second phase of RTMP, then RTMP would be able to achieve higher compression efficiency than IBC since a full-RD test would be applied for each BV. Further, in such a case, the encoder of RTMP would be much more complex than the encoder of IBC due to the same reason.

The above analysis is carried out for various sequences and the corresponding test results are summarized in Table I. Note that when $n_r = 1$, RTMP is identical to TMP. The general observation from the experimental results is that, for any sequence, the percentage of occurrence at $\Delta\text{RTMP}_x = 0, \Delta\text{RTMP}_y = 0$ increases as the value of n_r increases, i.e., the number of instances where the RTMP predictor is identical to the IBC predictor increases as its number of regions increases. Additionally, the distortion from RTMP decreases as the number of regions increases. Furthermore, in an ideal case, the RTMP distortion approaches the optimal (copy prediction) distortion when $n_r = n_s$. Hence, altogether, it is concluded that the displacement error from RTMP is lower than that from TMP, i.e.,

$$\Delta\text{RTMP} \leq \Delta\text{TMP} \tag{6}$$

and

$$D_{\text{RTMP}} \leq D_{\text{TMP}} \tag{7}$$

where ΔRTMP and ΔTMP are the displacement errors, and D_{RTMP} and D_{TMP} are the distortions from the RTMP and TMP methods respectively. In other words, the prediction efficiency of RTMP is higher than the corresponding efficiency of TMP.

Lastly, Table I also demonstrate the adaptability of RTMP between TMP and IBC by varying its number of regions n_r.

TABLE I

PERCENTAGE OF OCCURRENCE AT $\Delta\text{RTMP}_x = 0$, $\Delta\text{RTMP}_y = 0$ FOR DIFFERENT NUMBERS OF REGIONS n_r (WITH THE SAME REFERENCE AREA) OF INTRA-RTM AND CORRESPONDING AVERAGE DISTORTION USING AI CONFIGURATION WITH QUANTIZATION PARAMETER $Q_p = 22$ FOR VARIOUS SEQUENCES (1 FRAME) WHERE THE TEST IS RESTRICTED TO 4×4 BLOCKS

	n_r	Percentage of occurrence	Avg. distortion
Johnny	1 (TMP)	11.75%	3923
	3	14.58%	2547
	5	16.45%	2200
	9	19.31%	1965
	17	22.00%	1957
	...	...	...
	...	...	...
	n_s (IBC)	100%	1190
BasketBallPass	1 (TMP)	9.18%	4413
	3	11.37%	3027
	5	13.18%	2687
	9	15.97%	2377
	17	18.40%	2349
	...	...	...
	...	...	...
	n_s (IBC)	100%	1412
BQSquare	1 (TMP)	4.87%	26134
	3	6.75%	17590
	5	8.30%	15553
	9	10.10%	13612
	17	12.00%	13264
	...	...	...
	...	...	...
	n_s (IBC)	100%	6697
SlideShow	1 (TMP)	13.36%	47058
	3	15.51%	29746
	5	17.18%	26014
	9	19.55%	23467
	17	21.45%	23258
	...	...	...
	...	...	...
	n_s (IBC)	100%	8995

B. Rate-Distortion Performance of RTMP

In the previous subsection we showed that the prediction efficiency of RTMP is higher when compared to that of TMP. In this subsection, an expression for a simplified RD cost of RTMP is formulated and evaluated against the simplified RD cost of TMP.

Let D_m be the distortion between the original block and the block predicted through method m, and R_m the rate associated with the side information from method m, then the simplified RD cost related to method m is given by,

$$J_m = D_m + \lambda R_m. \tag{8}$$

Here, λ is the Lagrange parameter that determines the trade-off between D_m and R_m, and is set according to

$$\lambda = c \cdot Q^2 \tag{9}$$

where Q is the quantization step-size and c is a constant [2], [43], [44], [45]. Further, Q is defined as

$$Q = 2^{(Q_p - 4)/6 + \kappa - 8} \tag{10}$$

where Q_p is the QP and κ is the bit-depth in bits per sample [46].

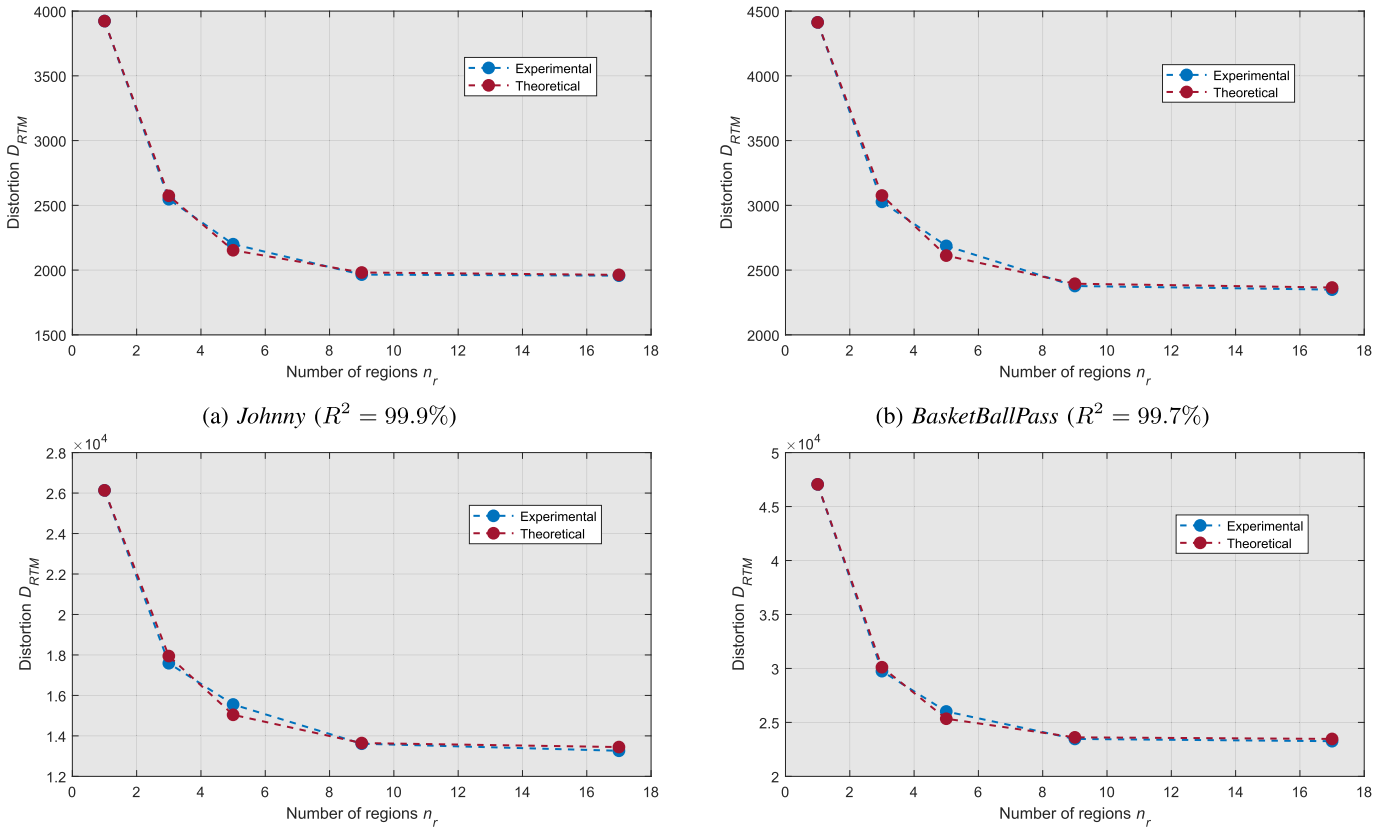
Now, for TMP, the rate associated with it is,

$$R_{\text{TMP}} = 0, \tag{11}$$

as there is no side information to be conveyed to the decoder. Hence, if D_{TMP} is the distortion related to TMP, then the simplified RD cost of TMP according to Eq. (8) is,

$$J_{\text{TMP}} = D_{\text{TMP}}. \tag{12}$$

The above expression indicates that the RD cost of TMP only depends on its distortion and remains constant over varying



(a) *Johnny* ($R^2 = 99.9\%$)

(b) *BasketBallPass* ($R^2 = 99.7\%$)

(c) *BQSquare* ($R^2 = 99.6\%$)

(d) *SlideShow* ($R^2 = 99.8\%$)

Fig. 2. Distortions of RTMP against number of region $n_r \in \{1, 3, 5, 9, 17\}$ for various sequences. R^2 gives the goodness-of-fit.

values of Q_p since the Lagrange parameter λ does not effect J_{TMP}. This is particularly advantageous for compression at low-rates (i.e., at high Q_p values).

Similarly, for the case of RTMP, suppose the reference area is partitioned into n_r regions and the region index is binary coded, then the approximated rate for RTMP is,

$$R_{\text{RTMP}} \approx \log_2(n_r). \tag{13}$$

Therefore, if D_{RTMP} is the distortion related to RTMP, then the simplified RD cost of RTMP according to Eq. (8) is,

$$J_{\text{RTMP}} \approx D_{\text{RTMP}} + \lambda \log_2(n_r). \tag{14}$$

Next, an expression for D_{RTMP} is required to evaluate the RD cost J_{RTMP} associated with RTMP. From the RTMP distortion values collected for different values of n_r in the experiments in subsection II-A, it is observed that the distortion of RTMP decreases non-linearly as the value of n_r increases. This behaviour can be compared to an exponential decay function, as given below:

$$A(t) = A_0 e^{-\gamma t} + C \tag{15}$$

where $A(t)$ is the final amount at time t, A_0 is the initial amount, γ is the decay constant (i.e., the constant that determines the rate of decay) and C is an offset.

In the case of RTMP, as shown in the experimental results in subsection II-A, the distortion of RTMP decreases as the number of regions n_r in the reference area increases. Thus, the distortion of RTMP for a given reference area is influenced by the number of contained regions, and therefore for RTMP, t in Eq. (15) can be associated by n_r. Further, when $n_r = n_s$, RTMP is comparable to IBC, i.e., the lower bound of the distortion from RTMP depends on the distortion from IBC (assuming IBC gives the optimal copy prediction distortion). This implies that the offset C can be given as,

$$C = D_{\text{off}} = D_{\text{IBC}} + \omega \tag{16}$$

where ω is an additional distortion that depends on the content and search algorithm. Ideally, $\omega = 0$.

Additionally, when $n_r = 1$, $D_{\text{RTMP}} = D_{\text{TMP}}$ since RTMP is equivalent to TMP in that case. This can be interpreted using Eq. (15) and (16) for $n_r = 1$ as,

$$D_{\text{TMP}} = A_0 e^{-\gamma} + D_{\text{off}}. \tag{17}$$

TABLE II

CONSTANTS RELATED TO THEORETICAL RTMP DISTORTION IN FIG. 2. D_{TMP}, D_{IBC}, γ AND ω ARE TMP DISTORTION, IBC DISTORTION, DECAY CONSTANT AND ADDITIONAL DISTORTION RESPECTIVELY

	D_{TMP}	D_{IBC}	γ	ω
Johnny	3923	1190	0.583197	773
BasketBallPass	4413	1412	0.528929	953
BQSquare	26134	6697	0.518248	6748
SlideShow	47058	8995	0.633459	14472

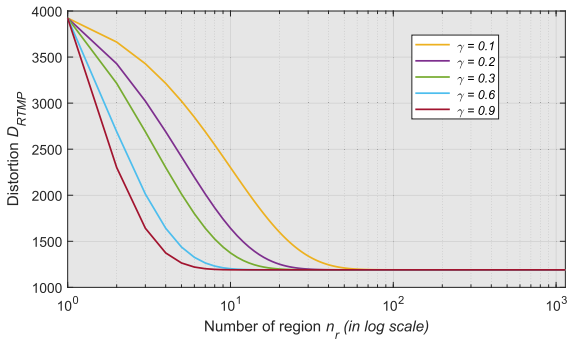


Fig. 3. Theoretical distortion of RTMP against varying number of regions n_r for decay constant $\gamma \in \{0.1, 0.2, 0.3, 0.6, 0.9\}$ (with the distortions assumed to be $D_{\text{TMP}} = 3923$, $D_{\text{IBC}} = 1190$, $\omega = 0$ and the number of samples in the reference area n_s to be 1140).

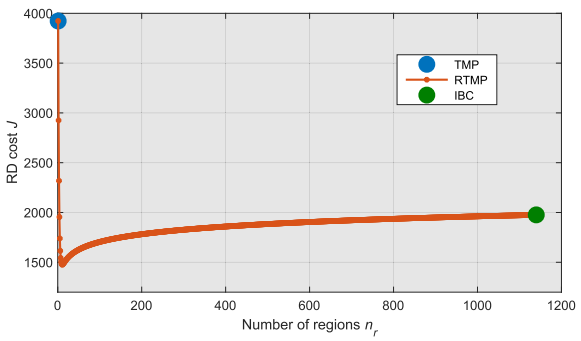


Fig. 4. Theoretical RD cost of RTMP against varying number of regions n_r for decay constant $\gamma = 0.5$ and quantization parameter $Q_p = 32$ (with the distortions assumed to be $D_{\text{TMP}} = 3923$, $D_{\text{IBC}} = 1190$, $\omega = 0$, and the number of samples in the reference area n_s to be 1140).

Hence,

$$A_0 = (D_{\text{TMP}} - D_{\text{off}})e^{\gamma}. \tag{18}$$

At last, substituting the above terms for A_0 and C in Eq. (15), the expression for the RTMP distortion can be approximated as,

$$\begin{aligned} D_{\text{RTMP}} &\approx (D_{\text{TMP}} - D_{\text{off}})e^{\gamma}e^{-\gamma n_r} + D_{\text{off}} \\ &\approx (D_{\text{TMP}} - D_{\text{off}})e^{-\gamma(n_r-1)} + D_{\text{off}}. \end{aligned} \tag{19}$$

Thus, the distortion D_{RTMP} of RTMP depends on the number of regions n_r and decay constant γ for a given value of the distortions D_{TMP}, and D_{off}.

The experimental and theoretical distortions of RTMP against number of regions $n_r \in \{1, 3, 5, 9, 17\}$ for the same sequences used in the experiments in subsection II-A are plotted in Fig. 2. The theoretical values are calculated using the constants (see Table II) that are collected through curve fitting of the corresponding experimental values. It can be observed that the theoretical values consistently approximate the experimentally obtained values.

The dependency of distortion D_{RTMP} of RTMP against the number of regions n_r for varying values of the decay constant γ is illustrated in Fig. 3. Since the experimentally obtained values of γ are between 0 and 1, the same range of values is considered for the theoretical analysis also. The distortions of TMP and IBC are assumed to be $D_{\text{TMP}} = 3923$ and $D_{\text{IBC}} = 1190$ respectively, and the number of samples in the reference area n_s to be 1140 (from the experiments related to the sequence *Johnny* in subsection II-A). It can be observed that D_{RTMP} decreases exponentially from the initial value D_{TMP} with increasing value of n_r and eventually equals D_{IBC}, provided $\omega = 0$. In other words, D_{RTMP} can be varied between D_{TMP} and D_{IBC} by simply modifying the number of regions n_r in the reference area. Besides, Fig. 3 show that D_{RTMP} decreases with increasing value of γ as expected from the general behaviour of an exponential decay function.

At last, applying Eq. (19) into Eq. (14) we obtain,

$$J_{\text{RTMP}} \approx [(D_{\text{TMP}} - D_{\text{off}})e^{-\gamma(n_r-1)} + D_{\text{off}}] + \lambda \log_2(n_r). \tag{20}$$

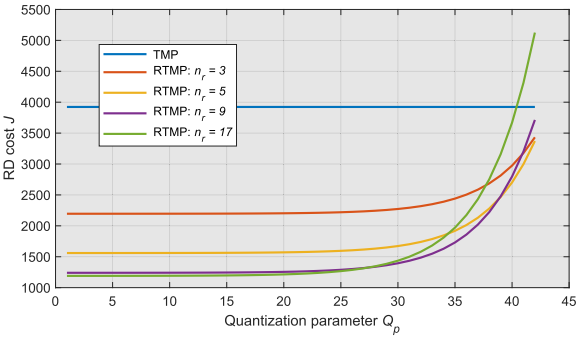


Fig. 5. Theoretical RD cost of RTMP (for decay constant $\gamma = 0.5$) and TMP against varying quantization parameter Q_p (with the distortions assumed to be $D_{\text{TMP}} = 3923$, $D_{\text{IBC}} = 1190$, $\omega = 0$, and the number of samples in the reference area n_s to be 1140).

The relationship between the RD cost J_{RTMP} and the number of regions n_r for a given QP and reference area (i.e., for a given value of γ, n_s, ω, D_{TMP} and D_{IBC}) using Eq. (20) is plotted in Fig. 4. The constant c and bit-depth κ for calculating the Lagrange parameter λ (see Eq. (9) and (10)) are assumed to be 0.12 and 8 respectively [45]. It is clear that the RD cost of RTMP is smaller than the related cost of TMP for the chosen QP and reference area. Further, as the value of n_r increases, the RD cost of RTMP rapidly decreases to a minimum value (say at $n_r = n_{rthres}$), and after that, slowly increases. This increase in the RD cost beyond n_{rthres} can be justified by the fact that the rate of signalling of the region index becomes substantially large for such instances, even though the corresponding distortion is rather small. Consequently, the value of n_r for the RTMP algorithm should be chosen such that $n_r \leq n_{rthres}$. Besides, as $n_{rthres} \ll n_s$, the value of n_r should also be selected such that $n_r \ll n_s$.

In order to further evaluate the behaviour of the RD cost of TMP and RTMP, their corresponding theoretical values are calculated using Eq. (12) and (20) respectively for $D_{\text{TMP}} = 3923$, $D_{\text{IBC}} = 1190$, $\omega = 0$, $n_s = 1140$, $c = 0.12$ and $\kappa = 8$, and plotted against the quantization parameter $Q_p \in \{1, 2, 3 \ldots 42\}$ as shown in Fig. 5. For the case of RTMP, the number of regions n_r considered for the examination are $n_r \in \{3, 5, 9, 17\}$, and the decay constant γ of D_{RTMP} is assumed to be 0.5. As expected, the RD cost of TMP remains constant over the varying values of Q_p. On the other hand, the

RD cost of RTMP for any value of n_r starts from a smaller value than that of TMP, remains constant until a particular value of Q_p (until $Q_p = Q_{pthres}$), and then, increases steadily. It is also observed that the rate of increase in the RD cost of RTMP for a smaller value of n_r is smaller than that with a higher value of n_r. Further, the RD cost of RTMP remains smaller than the corresponding cost of TMP for a wide range of Q_p values. More precisely, TMP wins over RTMP only at very high Q_p values as its speciality of no data overhead is particularly beneficial in such occasions. Furthermore, the RD cost for RTMP is supposedly smaller than in the given model, since entropy coding of the region index is not taken into account in the current analysis.

Apart from the findings mentioned above, it should be pointed out that the decoder complexity of RTMP is significantly smaller compared to the corresponding complexity of TMP. This is because the search routine for the predictor(s) in RTMP is executed only in the given region of the reference area instead of the complete reference area as in TMP. Further, this leads to reduced memory accesses in RTMP with respect to TMP at the decoder-side. Altogether, it can be concluded that higher coding and computational efficiencies than TMP can be achieved with the RTMP approach. In other words, RTMP outperforms TMP.

III. Implementation of Region-Based Template Matching Prediction in Block-Based Video Codecs

In this section, the application of the RTMP method for intra coding on a standard block-based video codec is explained.

A. Algorithm

Let B_c be the $M \times N$ block to be predicted and X be the current partially reconstructed picture. Then, the search window in the immediate neighbourhood of B_c is partitioned into n_r regions according to the constraints from (a) to (e), as illustrated in Fig. 6. Any region is represented by Γ_ν where the value of ν belongs to $\nu = 1, 2, 3, \ldots, n_r$.

(a) The borders of the regions are clearly defined so that the TM search for the predictor block at the decoder is synchronized with that at the encoder. Additionally, the regions are non-overlapping in order to prevent rechecking of the reference templates.

(b) The first region Γ_1 is covered by the top and left area of B_c. As mentioned before, the immediate neighbourhood of the current block is the *most probable region* and hence it is not partitioned.

(c) n_r is an odd number so that all the areas other than Γ_1 are partitioned with diagonal symmetry.

(d) $n_r - 1$ is a power of 2. This is for the efficient signalling of the region index ν (refer subsection III-E).

(e) As deduced from subsection II-B, the value of n_r is very small compared to the number of samples in the search window, i.e., $n_r \ll n_s$.

Assuming that the conditions from (a) to (e) are met, the relationship between the parameter that determines the size of the search window ζ and regions δ is generalized by,

$$\zeta = (\lfloor \frac{n_r}{2} \rfloor + 1) \cdot \delta \tag{21}$$

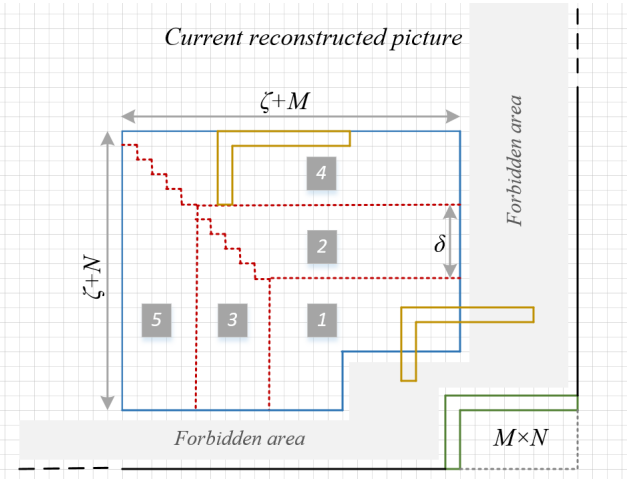


Fig. 6. Partitioning of the search window (marked by a solid blue line) into regions (number of regions $n_r = 5$) in intra-RTM for an $M \times N$ block where ζ is the search window size parameter and δ is the region size parameter. The current template is marked by a solid green line and examples of the reference templates are marked by solid yellow lines. The region index ν is highlighted by gray-encased numbers.

where $\lfloor \cdot \rfloor$ is the floor function.

Now in intra-RTM, given a region index ν, the TM search process is carried out in region Γ_ν for finding the predictor block of B_c. In detail, if τ_c is the template associated with B_c and τ_r^ν is any possible template in the region Γ_ν of X, then the SSD error between them is,

$$\epsilon_r^\nu = \mathrm{SSD}(\tau_c, \tau_r^\nu) = \sum_{j=1}^{n_\tau} (\tau_c(j) - \tau_r^\nu(j))^2 \tag{22}$$

where n_τ is the number of samples in the template that are used for error calculation and j corresponds to individual samples in the template. Then, the reference template τ_r^ν that gives the minimum error against τ_c is termed as the best template τ_b^ν from the region Γ_ν, i.e.,

$$\tau_b^\nu = \arg\min_r (\epsilon_r^\nu). \tag{23}$$

Finally, the block corresponding to τ_b^ν is regarded as the predictor B_p^ν of the current block B_c from Γ_ν.

Later, the prediction signal of RTMP from the given region is generated using the adaptive weighted averaging (AWA) technique. It is explained in the next subsection.

B. Adaptive Weighted Averaged Prediction

The prediction efficiency of a copy prediction method can be improved by using more than one predictor block. In that case, conventionally, the average of the corresponding samples of the predictors is used as the final prediction signal. The averaging process causes a smoothing effect (provided the multiple predictors have a similar distortion) and the distortion of the predicted block against the original block decreases, resulting in an improved outcome. Various publications on TMP in the literature have demonstrated that enabling averaging can enhance the coding efficiency of TMP (for examples see [21], [22], [23], [35]). Nevertheless, our studies (refer Appendix) indicated that the usage of a fixed number of predictors is not favourable in all cases. This observation has

led to the development of AWA. In AWA, the final number of predictor blocks and their weights in the prediction signal generation of RTMP is decided based on the TM error between the best predictor and other predictors. It is described in the following.

Let n_p be the number of predictors collected from the given region through TM search. Based on the detailed investigation on the value of n_p for intra-RTM on H.266/VVC, it is found that $n_p = 3$ is a suitable choice. Now, let P_1, P_2, P_3 are the multiple predictor blocks from the given region and $\epsilon_{p1}, \epsilon_{p2}, \epsilon_{p3}$ are the SSD error related to them respectively such that $\epsilon_{p1} \leq \epsilon_{p2} \leq \epsilon_{p3}$. Then, the final prediction of intra-RTM,

$$P_{final} = \begin{cases} \frac{(2P_1+P_2+P_3)}{4}, & \text{if } \epsilon_{p3} \leq \epsilon_{thres} \\ \frac{(P_1+P_2)}{2}, & \text{else if } \epsilon_{p2} \leq \epsilon_{thres} \\ P_1 & \text{otherwise} \end{cases} \tag{24}$$

where $\epsilon_{thres} = 2\epsilon_{p1}$. In this way, the final number of predictor blocks and their weights are not predetermined in RTMP, instead, they are decided at the end of the TM search algorithm based on the TM error between the best predictor (P_1) and other predictors.

Note that separate TM search routines are *not* initiated for finding the additional predictors when averaging is enabled, i.e., the given region is not searched multiple times but only once. All the n_p predictors are gathered in a single iteration. The major difference for the case of $n_p > 1$ is that a sorting algorithm is carried out at the end of every reference template check, in order to keep track of the best n_p predictors in the increasing order of their associated TM error. This may result in an increase in the complexity, especially if the value of n_p is large. Additionally, the multiple predictors have to be recorded and maintained during the complete TM search process.

C. Encoder Search

At the encoder-side, the RDO algorithm is utilized to identify the region that gives the minimum RD cost and also to determine whether RTMP is to be applied to the current block or not. In detail, individual prediction signal from each region is obtained one-by-one and their corresponding RD cost is calculated. Then, the region that gives the least cost among the n_r regions is considered as the best region. Later, the cost of the best region is compared with the cost of the other tools. Finally, if RTMP has the least cost among all, it is treated as the best mode of prediction for the current block B_c. In that case, the index ν of the best region is transmitted to the decoder in the bit-stream, along with the RTMP mode flag, for the final reconstruction of B_c.

D. Intra-RTM for Chroma

The intra-RTM method can be applied to chroma components in the same manner as luma. From the implementation perspective of H.266/VVC, there are two ways of doing this.

1) As an additional prediction mode - Here, intra-RTM is introduced as a separate chroma prediction mode. Thus, the chroma blocks would always have the option for coding through intra-RTM like the luma blocks. Consequently, supplementary full-RD tests (equivalent to the value of n_r) as in a luma block are required at the encoder, resulting in an increased encoder complexity. Further, extra bins are necessary for transmitting the region index of the chroma blocks to the decoder.
2) As an intra derived[3] mode (DM_CHROMA) - In this option, intra-RTM is applied to the chroma components only if the current chroma mode is DM_CHROMA and the corresponding luma mode is intra-RTM. Hence, no extra full-RD tests are required. Further, the value of the region index ν for the chroma components is copied from the corresponding luma component. Thus, there are no extra syntax elements for the chroma components.

Note that having the RTM tool *always* in the first option does not mean that the chroma block would be coded always with RTMP. This only implies that intra-RTM would be added always for full-RD testing, and based on the results, the final choice would be made, as in a luma block.

Since the second option offers a better gain-complexity trade-off, it is recommended for a more competent extension of intra-RTM to the chroma components. Note that this option is adopted for the chroma components in the experiments in section IV. Additionally, the region size parameter for the chroma predictors is modified as,

$$\delta_{chroma} = \lfloor \frac{\delta}{2} \rfloor \tag{25}$$

so that it is comparable to the typically used 4:2:0 chroma sub-sampling format. δ is the region size parameter of luma.

E. Signalling

In order to identify whether the RTMP method is to be applied to the current block or not, a flag (`intra_rtm_flag`) is added to every coding unit (CU). When it is true, the index ν of the region is decoded from the bit-stream.

The coding scheme related to the region index transmission is shown in Fig. 7. Since Γ_1 is the *most probable region*, a separate flag termed as the `mpr_flag` is utilized for this region. Therefore, in the region index coding of RTMP, the `mpr_flag` is decoded first. Suppose it is true, the value of ν is set to 1. On the other hand, if it is false, then the region index is parsed using fixed-length coding with a p number of bins. All the bins are coded using the context-adaptive binary arithmetic coding (CABAC) engine.

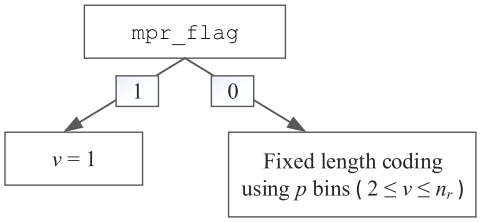


Fig. 7. Syntax elements of the region index coding.

IV. Experimental Results

The experimental results of the intra-RTM technique on top of the VVC Test Model (version VTM-14.0) are presented and discussed in this section. The JVET CTC defined

[3]In the intra derived mode, the intra prediction mode of the current chroma blocks is inherited from the corresponding luma block.

TABLE III

BD-RATE SAVINGS (Y) OF THE RTMP AND TMP METHODS FOR INTRA CODING AGAINST H.266/VVC (REFERENCE SOFTWARE VTM-14.0) FOR AI AND RA CONFIGURATIONS USING THE FOLLOWING SETTINGS WHERE η IS THE WIDTH OF THE TEMPLATE IN SAMPLES, ζ IS THE SEARCH WINDOW SIZE PARAMETER, δ IS THE REGION SIZE PARAMETER AND n_r IS THE NUMBER OF REGIONS. INTRA-RTM: $\eta = 1$, $\zeta = 60$, $\delta = 12$ AND $n_r = 9$ (FOR CLASSES F AND TGM, $\zeta = 60$, $\delta = 30$ AND $n_r = 3$). INTRA-TM: $\eta = 1$, $\zeta = 60$. ENCT AND DECT ARE THE ENCODER AND DECODER RUN-TIME RESPECTIVELY. AVG. IS THE AVERAGE BD-RATE SAVINGS OF CLASSES A1, A2, B, C AND E. TEST 1 - CODING TOOLS CONFIGURED AS IN THE JVET CTC. TEST 2 - TEST 1 WITH ISP, MIP, MRL DISABLED. TEST 3 - TEST 1 WITH IBC ENABLED

	Test 1				**Test** 2				**Test** 3			
Test	**Intra-TM**		**Intra-RTM**		**Intra-TM**		**Intra-RTM**		**Intra-TM**		**Intra-RTM**	
class	**AI**	**RA**	**AI**	**RA**	**AI**	**RA**	**AI**	**RA**	**AI**	**RA**	**AI**	**RA**
A1	-0.10%	0.01%	-0.14%	-0.02%	-0.13%	-0.02%	-0.20%	-0.06%	-0.06%	-0.02%	-0.09%	-0.03%
A2	-0.85%	-0.40%	-0.97%	-0.47%	-1.10%	-0.54%	-1.24%	-0.63%	-0.47%	-0.24%	-0.52%	-0.26%
B	-0.61%	-0.24%	-0.70%	-0.29%	-0.85%	-0.32%	-0.97%	-0.38%	-0.37%	-0.12%	-0.42%	-0.14%
C	-0.36%	-0.11%	-0.45%	-0.13%	-0.74%	-0.18%	-0.91%	-0.23%	-0.23%	-0.04%	-0.28%	-0.04%
E	-1.35%	—	-1.62%	—	-1.81%	—	-2.13%	—	-0.78%	—	-0.90%	—
Avg.	**-0.63%**	**-0.19%**	**-0.75%**	**-0.23%**	**-0.91%**	**-0.27%**	**-1.07%**	**-0.33%**	**-0.37%**	**-0.10%**	**-0.43%**	**-0.11%**
D	-0.36%	-0.09%	-0.37%	-0.07%	-0.53%	-0.08%	-0.65%	-0.08%	-0.10%	0.08%	-0.09%	0.08%
F	-0.92%	-0.86%	-1.00%	-0.89%	-1.31%	-0.96%	-1.37%	-0.99%	-0.92%	-0.86%	-1.00%	-0.89%
TGM	-3.43%	-1.98%	-3.57%	-2.01%	-4.38%	-2.13%	-4.50%	-2.15%	-3.43%	-1.98%	-3.57%	-2.01%
EncT	110%	103%	130%	107%	122%	103%	163%	107%	105%	102%	116%	106%
DecT	116%	104%	104%	99%	126%	104%	105%	99%	115%	100%	103%	101%

for SDR content is utilized for evaluating the test results [3], [42]. However, the tests are restricted for the all intra (AI) and random access (RA) configurations. Further, as recommended by the JVET CTC, the coding efficiency of a test is measured in Bjøntegaard-Delta (BD) bit-rate savings which represents the average bit-rate savings for the same video quality (here PSNR). It is denoted as a percentage of the reference bit-rate. Hence, if the measured BD-rate value results in a negative value, it implies a coding gain from the proposed method. On the contrary, a positive value of the same signifies coding loss [47]. Additionally, as per the JVET CTC, the tests are repeated for QP values $Q_p \in \{22, 27, 32, 37\}$ and their average BD-rate is considered as the final coding gain value of a test.

Three test results of intra-RTM are examined in this section. In the first test, all the VVC coding tools are configured as recommended by the JVET CTC. This helps to understand the performance of intra-RTM when all the VVC coding tools are enabled. However, the complete potential of intra-RTM would not be visible in this test due to the presence of other intra tools. Furthermore, all the coding tools in the standard are typically not enabled in a practical application. Hence, in the second test, only the basic intra tools are enabled (i.e., the intra tools ISP, MIP and MRL are disabled). At last, in the third test, IBC is enabled additional to the CTC. This test is intended to understand the performance of intra-RTM against IBC which is an established copy prediction method used for intra-picture prediction in screen content coding (SCC). The three above-mentioned tests are summarized below. Note that both the anchor and test are configured as given below; the only difference is that, in the test, intra-RTM is enabled.

- Test 1 - Coding tools configured as in the JVET CTC.
- Test 2 - Coding tools configured as in the JVET CTC, however, ISP, MIP and MRL are disabled.
- Test 3 - Coding tools configured as in the JVET CTC, however, IBC is enabled.

The corresponding results of the aforementioned tests are presented in Table III. The values of search window size parameter ζ, region size parameter δ and number of regions n_r related to intra-RTM are 60, 12 and 9 respectively (for the SCC sequences in classes F and TGM, they are 60, 30 and 3 respectively). Further, the template width η in samples is 1. Note that the values of ζ, δ, n_r, η and n_p are chosen experimentally such that they offer a reasonable trade-off between coding gain and computational complexity (refer Appendix). Additionally, for the purpose of comparison, the corresponding test results of intra-TM ($\zeta = 60, \eta = 1$) are included in Table III. The AWA option is applied to intra-TM also. However, in the standard TMP approach, normal or weighted averaging is typically used. Besides, similar to intra-RTM, intra-TM is enabled to the chroma components through the intra derived mode and the search window size parameter is modified as in Eq. (25).

An average BD-rate saving of -0.75% is obtained by intra-RTM in Test 1 with only 104% decoder run-time for AI configuration. For RA configuration, it is -0.23% with 99% decoder run-time. This test demonstrates the coding gain from intra-RTM when all the coding tools of H.266/VVC is enabled as per JVET CTC. The noticeable coding gains in all the classes indicate that intra-RTM is applicable to a wide variety of sequences. In Test 2, the average BD-rate savings is -1.07% with 105% decoder run-time. This test shows the potential of intra-RTM in a comparatively less competitive environment. In the case of Test 3, the average BD-rate savings from intra-RTM is -0.43% with 103% decoder run-time. This test indicates that intra-RTM can achieve a justifiable coding gain even in the presence of IBC. The comparatively less overhead than IBC has benefited intra-RTM in this test. Note that, according to JVET CTC, IBC is always enabled in classes F and TGM. That is why these classes have identical results in Tests 1 and 3.

Comparing the test results of intra-RTM and intra-TM, it is clear that intra-RTM has consistently higher BD-rate savings

TABLE IV

AVERAGE BD-RATE SAVINGS (Y) OF INTRA-RTM AGAINST VTM-14.0 FOR OPTIONS $O1$ TO $O6$ WITH TEMPLATE WIDTH $\eta = 1$, SEARCH WINDOW SIZE PARAMETER $\zeta = 60$, NUMBER OF REGIONS $n_r = 5$ AND REGION SIZE PARAMETER $\delta = 20$

	$O1$	$O2$	$O3$	$O4$	$O5$	$O6$
Avg.	-0.55%	-0.68%	-0.70%	-0.31%	**-0.72%**	-0.48%
EncT	121%	121%	121%	122%	**122%**	122%
DecT	103%	105%	106%	104%	**105%**	105%

TABLE V

OPTIONS FOR PREDICTOR AVERAGING ANALYSIS

	n_p	**AWA**	**weight of the predictor** P_1	P_2	P_3	P_4	**remarks**
$O1$	1	no	1	0	0	0	
$O2$	2	no	1	1	0	0	
$O3$	3	no	2	1	1	0	
$O4$	4	no	1	1	1	1	
$O5$	3	yes	2	1	1	0	if $\epsilon_3 \leq \epsilon_{thres}$
			1	1	0	0	else if $\epsilon_2 \leq \epsilon_{thres}$
			1	0	0	0	otherwise
$O6$	4	yes	1	1	1	1	if $\epsilon_4 \leq \epsilon_{thres}$
			2	1	1	0	else if $\epsilon_3 \leq \epsilon_{thres}$
			1	1	0	0	else if $\epsilon_2 \leq \epsilon_{thres}$
			1	0	0	0	otherwise

than intra-TM. This confirms the findings from section II that RTMP has higher coding efficiency than TMP. Besides, intra-RTM has negligible decoder run-time, almost as identical as the reference.[4] This is due to the fact that in RTMP the search routine for the predictors of the current block is initiated in the given region of the search window only whereas it is carried out in the entire search window in TMP. The experimental results further indicate that the bit-rate savings of intra-TM and intra-RTM are higher in AI configuration when compared to RA configuration. This is because intra-TM and intra-RTM are intra tools and are used predominantly in intra-pictures.

It is also noticed from the experimental results in Table III that the encoder run-time of intra-RTM is more than that from intra-TM. This is because the standard RDO mechanism is used to identify the best region in these tests. However, some optimizations as proposed in [38] and [39] can be adopted to considerably reduce the encoder complexity of RTMP. For example, when the optimizations were incorporated into a similar implementation of the given intra-RTM version on VTM-2.0.1, the encoder run-time reduced from 203% to 133% while the BD-rate savings dropped from -1.14% to -1.08%.

V. CONCLUSION

The RTMP technique is an advanced form of traditional TMP. In RTMP, the reference area is partitioned into multiple regions and the final prediction signal is obtained through a linear combination of the predictor blocks from the given region. The index of the chosen region is conveyed to the decoder in the bit-stream. In this publication, a detailed analysis of RTMP based on experimental data is presented. It is shown that the prediction efficiency from RTMP is higher than the associated efficiency from the standard TMP approach. Additionally, an example implementation of the RTMP method for intra coding on H.266/VVC is presented. The experimental results indicate significant coding gains against the H.266/VVC anchor. Further, the test results demonstrate that RTMP achieves more BD-rate savings than TMP. Besides, the decoder complexity of RTMP is much smaller when compared to the decoder complexity of standard TMP. Altogether, RTMP is more efficient than TMP, in terms of coding gain as well as computational complexity.

[4] A run-time value of 100% for the test, implies the same run-time as the reference.

TABLE VI

AVERAGE BD-RATE SAVINGS (Y) OF INTRA-RTM AGAINST VTM-14.0 FOR TEMPLATE WIDTH $\eta \in \{1, 2, 3, 4\}$ WITH NUMBER OF PREDICTORS $n_p = 1$, SEARCH WINDOW SIZE PARAMETER $\zeta = 60$, NUMBER OF REGIONS $n_r = 5$ AND REGION SIZE PARAMETER $\delta = 20$

	η **1**	**2**	**3**	**4**
Avg.	**-0.55%**	-0.60%	-0.59%	-0.57%
EncT	**121%**	127%	132%	139%
DecT	**103%**	106%	106%	108%

TABLE VII

AVERAGE BD-RATE SAVINGS (Y) OF INTRA-RTM AGAINST VTM-14.0 FOR SEARCH WINDOW SIZE PARAMETER $\zeta \in \{15, 30, 45, 60, 75, 90, 105, 120\}$ WITH NUMBER OF PREDICTORS $n_p = 1$, NUMBER OF REGIONS $n_r = 5$ AND TEMPLATE WIDTH $\eta = 1$

	ζ **15**	**30**	**45**	**60**	**75**	**90**	**105**	**120**
Avg.	-0.36%	-0.48%	-0.53%	**-0.55%**	-0.56%	-0.57%	-0.58%	-0.58%
EncT	111%	114%	118%	**121%**	127%	132%	137%	143%
DecT	102%	102%	103%	**103%**	104%	105%	109%	109%

APPENDIX

SUPPLEMENTARY EXPERIMENTAL RESULTS

Test results on various RTMP-parameter analyses are summarized here. All tests are with the number of regions $n_r = 5$.

1) Predictor averaging: The different predictor options considered for this analysis are given in Table V where $\epsilon_{thres} = 2\epsilon_{p1}$ (refer subsection III-B for more details on AWA and ϵ_{thres}). The value of the weights of the predictors is a power of 2 and the summation of all the weights is also a power of 2 such that multiplications and divisions can be efficiently implemented as bit shifts. The corresponding test results are in Table IV.
2) Template width η: The test results from this analysis with $\eta \in \{1, 2, 3, 4\}$ are given in Table VI.
3) Search window size parameter ζ: The test results from this analysis with $\zeta \in \{15, 30, 45, 60, 75, 90, 105, 120\}$ are given in Table VII.

REFERENCES

[1] R. S. Choras, Ed., "20 years of progress in video compression—From MPEG-1 to MPEG-H HEVC. General view on the path of video coding development," in *Image Processing and Communications Challenges 8*. Springer, 2017. [Online]. Available: https://www.researchgate.net/publication/310494503, doi: 10.1007/978-3-319-47274-4_1.

[2] V. Sze, M. Budagavi, and G. J. Sullivan, *High Efficiency Video Coding (HEVC) Algorithms and Architectures*. Springer, 2014. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-06895-4

[3] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 14 (VTM 14)*, document JVET-W2002, Jul. 2021.

[4] B. Bross et al., "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.

[5] T. Nguyen et al., "Overview of the screen content support in VVC: Applications, coding tools, and performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3801–3817, Oct. 2021.

[6] X. Xu, X. Li, and S. Liu, "Intra block copy for next generation video coding," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, San Diego, CA, USA, Jul. 2018, pp. 1–4.

[7] X. Xu et al., "Intra block copy in HEVC screen content coding extensions," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.

[8] C. Pang, J. Sole, Y. Chen, V. Seregin, and M. Karczewicz, "Intra block copy for HEVC screen content coding," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2015, p. 465.

[9] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.

[10] J. Li et al., "Intra block copy for screen content in the emerging AV1 video codec," in *Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Mar. 2018, pp. 355–364.

[11] Z. Zhang and V. Sze, "Rotate intra block copy for still image coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Quebec City, QC, Canada, Sep. 2015, pp. 4102–4106.

[12] J. R. Ohm, *Multimedia Signal Coding and Transmission*. Springer, 2016. [Online]. Available: https://link.springer.com/book/10.1007/978-3-662-46691-9

[13] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by template matching," in *Proc. Int. Conf. Image Process.*, Altlanta, GA, USA, Oct. 2006, pp. 1693–1696.

[14] Y. Guo, Y.-K. Wang, and H. Li, "Priority-based template matching intra prediction," in *Proc. IEEE Int. Conf. Multimedia Expo*, Hannover, Germany, Jun. 2008, pp. 1117–1120.

[15] Y. Zheng, P. Yin, O. D. Escoda, X. Li, and C. Gomila, "Intra prediction using template matching with adaptive illumination compensation," in *Proc. 15th IEEE Int. Conf. Image Process.*, San Diego, CA, USA, Oct. 2008, pp. 125–128.

[16] M. Moinard, I. Amonou, P. Duhamel, and P. Brault, "A set of template matching predictors for intra video coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2010, pp. 1422–1425.

[17] S. Cherigui, C. Guillemot, D. Thoreau, P. Guillotel, and P. Perez, "Hybrid template and block matching algorithm for image intra prediction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 781–784.

[18] T. Zhang, H. Chen, M.-T. Sun, D. Zhao, and W. Gao, "Hybrid angular intra/template matching prediction for HEVC intra coding," in *Proc. Visual Commun. Image Process. (VCIP)*, Singapore, Dec. 2015, pp. 1–4.

[19] H. Zhang et al., "Rotational weighted averaged template matching for intra prediction," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Bangkok, Thailand, Nov. 2019, pp. 373–376.

[20] M. Lei, F. Luo, X. Zhang, S. Wang, and S. Ma, "Two-step progressive intra prediction for versatile video coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1137–1141.

[21] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by averaged template matching predictors," in *Proc. CCNC*, Las Vegas, NV, USA, Jan. 2007, pp. 405–409.

[22] C. Lan, J. Xu, F. Wu, and G. Shi, "Intra frame coding with template matching prediction and adaptive transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 1221–1224.

[23] Y. Zhang, Y. Sun, Q. Zhang, and L. Yu, "Adaptive weighted averaged template matching prediction for intra coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, May 2018, pp. 1–4.

[24] K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, and C. S. Boon, "Inter frame coding with template matching spatio-temporal prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Singapore, Oct. 2004, pp. 465–468.

[25] Y. Suzuki, C. S. Boon, and T. K. Tan, "Inter frame coding with template matching averaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, San Antonio, TX, USA, Oct. 2007, pp. III-409–III-412.

[26] S. Kamp, M. Evertz, and M. Wien, "Decoder side motion vector derivation for inter frame video coding," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, San Diego, CA, USA, Oct. 2008, pp. 1120–1123.

[27] S. Kamp, B. Bross, and M. Wien, "Fast decoder side motion vector derivation for inter frame video coding," in *Proc. Picture Coding Symp. (PCS)*, Chicago, IL, USA, May 2009, pp. 1–4.

[28] S. Kamp and M. Wien, "Decoder-side motion vector derivation for hybrid video inter coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Singapore, Jul. 2010, pp. 1277–1280.

[29] R. Wang, L. Huo, S. Ma, and W. Gao, "Combining template matching and block motion compensation for video coding," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS)*, Dec. 2010, pp. 1–4.

[30] S. Kamp and M. Wien, "Decoder-side motion vector derivation for block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1732–1745, Dec. 2012.

[31] T. Chen, X. Sun, F. Wu, and G. Shi, "Adaptive patch matching for motion compensated prediction," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 2621–2624.

[32] C.-C. Chen, W.-H. Peng, and S.-C. Chou, "Multi-hypothesis temporal prediction using template matching prediction and block motion compensation for high efficiency video coding," in *Proc. 13th Pacific-Rim Conf. Adv. Multimedia Inf. Process.*, vol. 7674, Dec. 2012, pp. 164–175.

[33] W. Peng and C. Chen, "An interframe prediction technique combining template matching prediction and block-motion compensation for high-efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1432–1446, Aug. 2013.

[34] D.-J. Won and J.-H. Moon, "Advanced template matching prediction using a motion boundary," in *Proc. 2nd Int. Conf. Image Graph. Process.*, Feb. 2019, pp. 110–113.

[35] S. Kamp, J. Ballé, and M. Wien, "Multihypothesis prediction using decoder side-motion vector derivation in inter-frame video coding," in *Proc. SPIE*, vol. 7257, Jan. 2009, Art. no. 725704.

[36] G. Venugopal, P. Merkle, D. Marpe, and T. Wiegand, "Fast template matching for intra prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 1692–1696.

[37] G. Venugopal, D. Marpe, and T. Wiegand, "Region-based template matching for decoder-side motion vector derivation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Taiwan, Dec. 2018, pp. 1–4.

[38] G. Venugopal, P. Helle, K. Mueller, D. Marpe, and T. Wiegand, "Hardware-friendly intra region-based template matching for VVC," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Mar. 2019, p. 606.

[39] G. Venugopal, K. Müller, D. Marpe, and T. Wiegand, "A unified region-based template matching approach for intra and inter prediction in VVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taiwan, Sep. 2019, pp. 4115–4119.

[40] G. Venugopal, "Region-based template matching for next generation video coding," Ph.D. thesis, Fac. IV Elect. Eng. Comput. Sci., Technische Universität Berlin, Berlin, Germany, 2022.

[41] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 2 (VTM 2)*, document JVET-K1002, Ljubljana, Slovenia, Jul. 2018.

[42] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET Common Test Conditions and Software Reference Configurations for SDR Video*, document JVET-N1010, Geneva, Switzerland, Mar. 2019.

[43] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[44] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[45] I. H. Schwarz, "Image and video coding (SS 21)," FU Berlin, Berlin, Germany, Lect. Notes. Accessed: Nov. 10, 2021. [Online]. Available: http://www.inf.fu-berlin.de/lehre/SS21/ImageVideoCoding/ivc.htm

[46] H. Schwarz et al., "Quantization and entropy coding in the versatile video coding (VVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3891–3906, Oct. 2021.

[47] T. K. Tan et al., "Video quality evaluation methodology and verification testing of HEVC compression performance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 76–90, Jan. 2016.

Gayathri Venugopal received the M.Sc. degree in electrical and electronics engineering from the University of Plymouth, U.K., in 2015, and the Dr.-Ing. degree in electrical engineering and computer science from the Technical University of Berlin, Germany, in 2022.

She joined Fraunhofer Heinrich Hertz Institute (HHI), Berlin, Germany, in 2015, as a Marie Curie Fellow of the EU-funded Project, PROVISION ITN. Since 2017, she has been a Research Associate with the Video Coding Technologies Group, HHI. She has actively participated in the standardization process of H.266/VVC. She is interested in research related to machine learning, and image and video coding.

Karsten Müller (Senior Member, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the Technical University of Berlin, Germany, in 1997 and 2006, respectively. He has been with the Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, Berlin, since 1997, where he is currently the Head of the Efficient Deep Learning Group. His research interests include multi-dimensional video coding, efficient federated learning, and compression of neural networks. He has been involved in international standardization activities, successfully contributing to the ISO/IEC Moving Picture Experts Group for work items on visual media content description, multiview, multi-texture, 3D video coding, and neural network coding. He co-chaired an Ad Hoc Group on 3D Video Coding from 2003 to 2012. He has served as the chair in IEEE conferences and a Senior Area Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Jonathan Pfaff received the Diploma and Dr.rer.nat. degrees in mathematics from Bonn University, Bonn, Germany, in 2010 and 2012, respectively. After a Postdoctoral Research stay at Stanford University, he joined the Video Coding and Analytics Department, Heinrich Hertz Institute, Berlin, Germany, in 2015, where he is currently the Head of the Research Group on Video Coding Technologies, since 2020. He has successfully contributed to the efforts of the ITU-T Video Coding Experts Group in developing the Versatile Video Coding standard since 2017. His current research interests include image and video coding and machine learning.

Heiko Schwarz received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 1996 and 2000, respectively.

In 1999, he joined the Fraunhofer Heinrich Hertz Institute, Berlin, Germany. Since 2010, he has been the Head of the Research Group Video Coding Technologies (formerly, Image and Video Coding), Fraunhofer Heinrich Hertz Institute. In October 2017, he additionally became a Professor for Image Processing at the Free University of Berlin (FU Berlin). He has actively participated in the standardization activities of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group. He successfully contributed to the video coding standards H.264/AVC, H.265/HEVC, and H.266/VVC. He co-chaired various ad hoc groups of the standardization bodies and coordinated core experiments. He served as reviewer for various international journals and international conferences. He was appointed as a Co-Editor of H.264/AVC and as a Software Coordinator for the SVC reference software. From 2016 to 2019, he was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Detlev Marpe (Fellow, IEEE) received the Dipl.-Math. degree (Hons.) from the Technical University of Berlin, Berlin, Germany, in 1990, and the Dr.-Ing. degree from the University of Rostock, Rostock, Germany, in 2004.

In 1999, he joined the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, where he is currently the Head of the Image and Video Coding Group and the Video Coding and Analytics Department. He was a major Technical Contributor to the entire process of development of the H.264/MPEG-4 Advanced Video Coding (AVC) standard and the H.265/MPEG High Efficiency Video Coding (HEVC) standard, including several generations of major enhancement extensions. In addition to the CABAC contributions for both standards, he particularly contributed to the fidelity range extensions (which include the high profile that received the Emmy Award in 2008) and the scalable video coding extensions of H.264/MPEG-4 AVC. During the development of its successor H.265/MPEGHEVC, he also successfully contributed to the first model of the corresponding standardization project and further refinements. He made successful proposals to the standardization of its range extensions and 3D extensions. He has authored numerous publications in the research area of image and video coding, and holds several hundreds of internationally issued patents and patent applications in this area. His current research interests include still image and video coding, signal processing for communications and computer vision, machine learning, and information theory.

Dr. Marpe is a member of the Informationstechnische Gesellschaft of the Verband der Elektrotechnik Elektronik Informationstechnik e.V. He received several best paper awards for his publications. He was a recipient of the Karl Heinz Beckurts Award in 2011 and the Joseph von Fraunhofer Prize in 2004. He was a co-recipient of three Technical Emmy Awards as a Key Contributor and a Co-Editor of the H.264/MPEG-4 AVC standard in 2008 and 2009, respectively, and a Key Contributor of H.265/MPEG-HEVC in 2017. In recognition of his dedicated contributions and excellent management of the review process, the 2016 Best Associate Editor Award of the IEEE Circuits and Systems Society. From 2014 to 2018, he served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Thomas Wiegand (Fellow, IEEE) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Hamburg-Harburg, Germany, in 1995, and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000.

As a student, he was a Visiting Researcher at Kobe University, Japan, and the University of California at Santa Barbara and Stanford University, USA, where he also returned as a Visiting Professor. He was a consultant and co-founder of several start-up companies. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Technical University of Berlin. He is jointly the Head of the Fraunhofer Heinrich Hertz Institute, Berlin, Germany. Since 1995, he has been an active participant in standardization for multimedia with many successful submissions to ITU-T and ISO/IEC. For his research, he received numerous research and innovation awards and various best paper awards for his publications. Since 2014, Thomson Reuters named him in their list of "The World's Most Influential Scientific Minds" as one of the most cited researchers in his field. He has been elected to the German National Academy of Engineering (Acatech) and the National Academy of Science (Leopoldina). Since 2018, he has been appointed the Chair of the ITU/WHO Focus Group on Artificial Intelligence for Health.