The CSI Journal on
Computer Science and Engineering
Vol. 18, No. 1, 2020
Pages 20-27
Regular Research Paper

A Novel Approach to Improve Rate-Distortion-Complexity in Versatile Video Coding Standard

Amir Rezaeieh[1] and Hoda Roodaki[1,2]

[1]Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran
[2]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Abstract

Versatile Video Coding (VVC) achieves up to 30% bitrate reduction at the same quality level compared to its predecessor, High Efficiency Video Coding (HEVC). It could support resolutions from 4K to 16K as well as 360° videos. Some new coding tools, such as AFFINE, Integer Motion Vector (IMV), Decoder-side Motion Vector Refinement (DMVR), and Triangle are proposed for VVC to improve the encoder efficiency. But, these new coding tools usually impose high computational complexity on the encoder side. In this paper, we provide a new approach to reduce the computational complexity of the Rate-Distortion Optimization (RDO) process in the encoder side of VVC. In the proposed approach, first, the effectiveness of each coding tool at various parts of the scene is estimated. The results of the experiments show that some of the coding tools--,i.e., AFFINE and IMV, have much better performance in borderline CTUs. So, the proposed approach suggests considering these coding tools in the RDO process, just for the borderline CTUs. This way the computational complexity is decreased considerably without affecting the coding performance. Simulation results show that by disabling the AFFINE and IMV coding tools in the rate-distortion optimization process of non-borderline CTUs, the encoding gain is reduced by only 0.88% and 0.72% BD-rate, but the processing time is reduced by 11.70% and 63.91%, respectively. As the second approach, the correlation between the various coding tools is investigated. Our simulation results show that the AFFINE and Triangle coding tools are highly correlated to each other. So, in the rate-distortion process, if the encoder decided to disable the AFFINE coding tool, the Triangle coding tool is also can be considered disabled without examining the rate-distortion process for this coding tool. This way, the computational complexity is reduced, by 4.96%, on average, without affecting the encoding gain considerably.

Keywords: Versatile Video Coding standard (VVC), AFFINE coding tool, DMVR coding tool, GBI coding tool, BIO coding tool, Triangle coding tool, IMV coding tool.

1. Introduction

HDTV (High-Definition Television) has become popular due to its higher resolution than traditional television systems. The data rate of uncompressed HD video can be as high as 3.0 Gbps and the bitrate is expected to rise with increases in resolution and color depth [1]. With the advent of 8K UHD commercial broadcasting services, the amount of video data that should be sent over communication networks has increased significantly. As the upcoming standard, Versatile Video Coding (VVC) achieves up to 30% bitrate reduction at the same quality level compared to High-Efficiency Video Coding (HEVC) using the most advanced video coding technologies and could be well adapted to high-resolution video [2]. VVC employs block-based video coding, such as the previous standards, in which, the pictures are divided into some blocks and each of them is either temporally or spatially predicted from the adjacent frames or the adjacent blocks in the current frame. The next steps are transform coding of prediction residues, quantization of transform coefficients, and entropy coding.

To improve the coding efficiency of VVC compared to the state-of-the-art video coding standards such as HEVC, some new coding tools are introduced for VVC that some of them are reviewed in the next section. The main problem of using these coding tools is their high computational complexity. In the Rate-Distortion Optimization (RDO) process, the performance of each coding tool is investigated for each Coding Tree Unit (CTU). Then, the coding tools that improve the coding efficiency of each CTU are selected for the encoding process of that CTU. This way, investigating the amount of coding efficiency of each coding tool for each CTU is computationally complex. To make this video coding standard more applicable, its computational complexity should be reduced extensively.

This paper is an extension of our recent work [3] that provides two new approaches to reduce the computational complexity of the rate-distortion process in the encoder of VVC. In the first approach, the effectiveness of each coding tool at various parts of the scene is examined. The results of experiments show that some of these coding tools have much better performance in borderline CTUs. So, we have considered them in the rate-distortion optimization process, just for the borderline CTUs to reduce the computational complexity of the rate-distortion process. This way, the coding performance is not affected extensively. In the second approach, the coding performance of various coding tools is investigated in relation to each other. The results of the observations show that some of the coding tools are highly correlated to each other in terms of compression efficiency. This correlation can be used to reduce the computational complexity of the encoder side.

The rest of the paper is organized as follows. Section 2 reviews the relevant background of the VVC standard and related coding tools. The related works are reviewed in Section 3. Section 0 presents the proposed method. Section 5 provides the experimental results, and Section 6 concludes the paper.

2. Background

2.1. VVC encoding process overview

The main part of the VVC standard is CTU partitioning in which, a frame is split into some CTUs for encoding. The maximum supported luminance block size in VVC is 128×128 and each CTU is divided into some Coding Units (CU) of different depths according to the quadtree plus a nested multi-type tree partitioning [4]. The maximum and minimum allowed CU size, which is 64×64 and 8×8, can be used to improve the coding efficiency in a smooth and complex region, respectively, as shown in Figure 1. In the prediction process, each CU is divided into one or more Prediction Units (PU), and in the transformation process, the CU will be divided into one or more Transform Units (TU) [4].

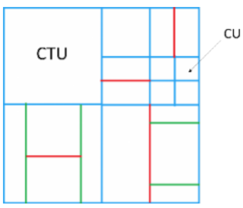

Figure 1: An Example of CTU partitioning in VVC standard

2.2. VVC coding tools

2.2.1. AFFINE coding tool

The complex motions, such as rotation and zooming, in natural videos, cannot be described by the translational motion model of traditional video coding standards. The high-order motion models, such as AFFINE, are proposed to characterize these complex motions [5] as shown in Figure 2.

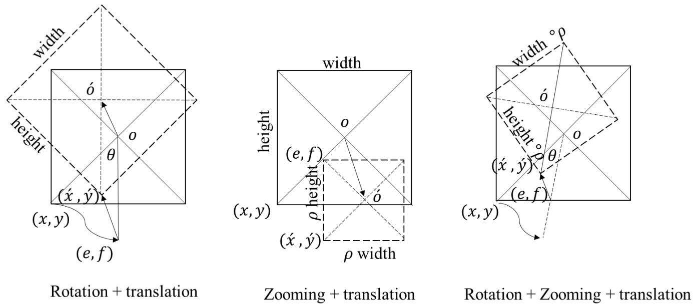

Figure 2: Examples of the affine model [5]

The combination of rotation and translation, the combination of zooming and translation, and the combination of rotation, zooming, and translation model are described using Eq. (1), (2), and (3), respectively [5].

$$\acute{x} = \cos\theta \times x + \sin\theta \times x + c$$
$$\acute{y} = -\sin\theta \times y + \cos\theta \times y + f \quad (1)$$

$$\acute{x} = \rho \times x + c$$
$$\acute{y} = \rho \times y + f \quad (2)$$

$$\acute{x} = \rho\cos\theta \times x + \rho\sin\theta \times y + c$$
$$\acute{y} = -\rho\sin\theta \times y + \rho\cos\theta \times y + f \quad (3)$$

where (x, y) is the bottom-left coordinate of the original block and $(\acute{x}, \acute{y})$ is the bottom-left coordinate of the new block with rotation, zooming, and translation. The parameter θ shows the rotation angle, c and f are the translation motion parameters, and ρ is the zooming factor in both x and y directions [5].

2.2.2. Integer motion vector (IMV)

The motion estimation in traditional video codecs is limited to integer-pixel accuracy. However, a moving object could be moved to a position that is not on the pixel grid and is between the pixels. So, the recent video codecs allow half-pixel accuracy motion estimation to increase the accuracy of prediction. But, since, the interpolation should be done before the motion estimation process to produce the pixel values at the half-pixel positions as shown in Figure 3, the computational complexity of the motion estimation process could be increased. Due to this high computational complexity, the integer motion estimation and the corresponding Integer Motion Vector (IMV) are still valuable in recent video codecs extensively to decrease the computational complexity [6].

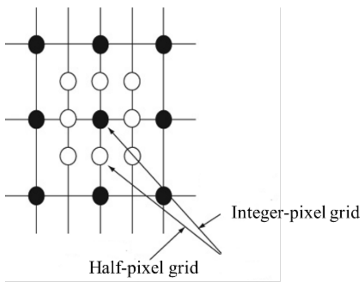

Figure 3: Integer pixels and half pixels in the motion estimation process

2.2.3. Decoder-side motion vector refinement (DMVR)

This coding tool is used in the bi-prediction motion estimation process to rectify the extracted motion vectors to make them more accurate.

In the VVC standard, due to the use of B-picture, the bi-prediction blocks in a B-picture allow us to use an arbitrary set of reference pictures in forward and backward directions, which is named list1 and list0, respectively [14].

So, for each coding block, two prediction blocks are selected using the motion vectors of list0 and list1 [7]. The DMVR coding tool is used to refine the extracted motion vectors using a weighted combination of the two prediction blocks extracted from the initial motion vectors, MV0 of list0 and MV1 of list1, as shown in Figure 4.

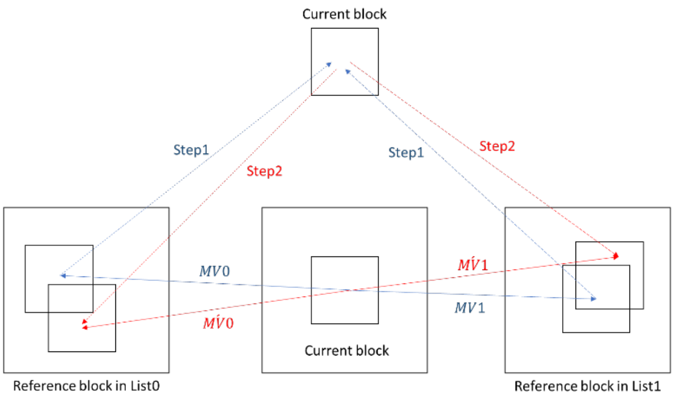

Figure 4: DMVR process to refine the motion vectors [7]

First, two prediction blocks are extracted, using two motion vectors, MV0 and MV1. A template block is the average of these two blocks. Then, the blocks at eight neighboring pixels around the original motion vectors are compared with the template block. The positions with a minimum cost are specified as MV0′ and MV1′. Finally, these updated motion vectors are used to generate the final predicted blocks.

2.2.4. Generalized BI-prediction (GBI)

Bi-directional motion compensated prediction is used to remove the temporal redundancy using the temporal correlation between pictures. The average of two uni-prediction signals using a weight value equal to 0.5 is used for this tool [8]. GBI is a generalized version of bi-directional prediction that suggests not to restrict the weight values to 0.5. In addition, the weight values may be different from a block to another as shown in EQ. 4 [8].

$$P[x] = (1 - w) \times P0[x + v0] + w \times P1[x + v1] \qquad (4)$$

where P[x] is the prediction of the current block located at a picture position x, P0[x+v0], is a motion-compensated prediction of the current block using a motion vector v0 from a reference picture in reference list L0. P1[x+v1], is a motion-compensated prediction of the current block using a motion vector v1 from a reference picture in reference list L0. 1-w and w are the weight values applied respectively to P0[x+v0] and P1[x+v1].

2.2.5. Bi-Directional optical flow (BIO)

In conventional video coding, the concept of B-picture is used to predict a block from two previously encoded frames, one from the reference frame in past and the other from the reference frame in the future to make the prediction of the current block more accurate. However, even using this procedure, some motions may remain in some parts of the block. Since the size of these regions may be smaller than the smallest acceptable prediction block size. So, it is desirable to have some way to compensate motion for every pixel. Then, to decrease the total bitrate, the motion vector signaling should be done only for the entire block. BIO is used to refine the pixel-like motion without any requirement to send additional signals to the decoder [9].

2.2.6. Triangle partition mode

In the VVC standard, the Triangle Partitioning Mode (TPM) is proposed for more flexible inter prediction compared to the state-of-the-art video coding standards. This mode tries to divide a rectangular coding block into two triangle prediction blocks as shown in Figure 5. Each triangular part has its motion vector (MV) [10] that leads to some BD-rate saving and increment in encoding time [11].

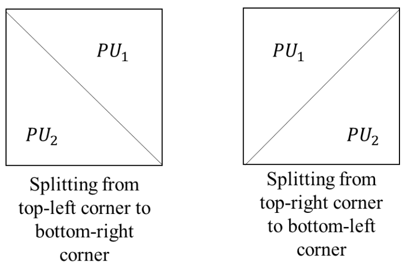

Figure 5 : Splitting a CU into two triangular prediction units [10]

2.2.7. Rate-distortion optimization process

The Rate-distortion optimization process in VVC is used to select the best coding tools for each CTU in the encoding process according to the rate and distortion cost [12]. The bitrate cost R and the distortion cost D are combined in a single cost function J as shown in equation (5) [13].

$$J = D + \lambda \times R \qquad (5)$$

Then, the RDO process is used to find a coding tool that minimizes this cost function.

3. Related work

In this section, the most recent methods suggested for reducing the computational complexity at the encoder side in the VVC standard are reviewed. In [14], the effect of some parameters such as special resolution and Quantization Parameter (QP) on encoder complexity is investigated. The proposed approach tries to change the CTU partitioning, intra mode, and transform selection to reduce the computational complexity.

In [15], a set of coding modes that can be disabled to preserve the coding efficiency is proposed. It is investigated that the joint disabling of some coding tools, such as AFFINE, AMVR, and GBI leads to complexity reduction considerably, while the compression efficiency is not changed significantly.

A low complexity DMVR scheme for inter prediction in VVC is presented in [16] that uses the integer-based DMVR to eliminate the intermediate interpolation filters. In addition, the DMVR is disabled for small blocks. In [17], the information of the parent CUs in multiple-type tree (MTT) segmentation structure and the number of reference frames are considered to reduce the computational complexity of the AFFINE motion estimation process. A method to reduce the computational complexity of DCT and DST processes by reducing the number of arithmetic operations is presented in [18]. On [19] side information is used to control the complexity of distributed scalable video coding. Finally, [20] proposes a CU depth prediction method to reduce the number of CU executions to decrease the rate-distortion costs of the adjacent CUs.

The rate-distortion process is computationally extensive, since, a large number of coding tools need to be considered in this process. The proposed method in this paper tries to reduce the number of coding tools that are considered in this process according to the video content and the correlation between the coding tools.

4. Proposed method

In this section, we introduce our proposed approaches to reduce the computational complexity of the rate-distortion process in VVC. In each part, we start by explaining the basic observations that have led to our methods.

4.1. The first approach: evaluation of the performance of various coding tools spatially

4.1.1. Observations

In this section, the efficiency of various coding tools in VVC standard in various parts of the scene are examined to show the effectiveness of various coding tools at different parts of the scene. For this purpose, various test sequences [21] with different resolutions, were selected and their properties are summarized in Table 1.

The VTM reference software version 4.0 [22] is used to extract the result and the coding performance of six different coding tools, i.e., GBI, BIO, DMVR, AFFINE, IMV, and Triangle in various parts of the scene are investigated. First, these coding tools are enabled using the “encoder_random_access” configuration file. Then, two Groups of Pictures (GoP) of all of the test sequences are encoded using four different QP values, 22, 27, 32, and 37 and the bitrate and PSNR of all CTUs in each frame are extracted.

Table 1. The properties of test sequences [21]

Test sequences	Resolution	Frame rate
BasketBallDrive	1920×1080	50
BQTerrace	1920×1080	60
Cactus	1920×1080	50
FourPeople	1280×720	60
Johnny	1280×720	60
KristenAndSara	1280×720	60
BasketballDrill	832×480	50
BQMall	832×480	60
PartyScene	832×480	50
RaceHorses_big	832×480	30

Then, each coding tool is disabled one by one and the test sequences are encoded with four QP values to extract the bitrate and PSNR of various CTUs. The Bjøntegaard-Delta bitrate (BD-bitrate) [23] is used to measure the Rate-Distortion (RD) performance when each coding tool is enabled versus when it is disabled. The results are shown in Table 2. We can see the performance of each coding tool for the borderline CTUs when the corresponding coding tool is enabled versus when it is disabled. The positive numbers show the gain when each coding tool is enabled. As we can see, for some specific coding tools, i.e., AFFINE and IMV, the main coding performance is in specific parts of the scene which is the borderline CTUs.

Table 2. The BD-rate of the borderline and central CTUs when each coding tool is enabled versus when it is disabled, for all test sequences, on average.

Coding Tools	BD-rate of the borderline CTUs	BD-rate of the central CTUs	Difference of BD-rates of borderline and central CTUs
AFFINE	6.74%	3.08%	3.66%
DMVR	4.25%	7.74%	-3.49%
BIO	3.17%	2.26%	0.91%
GBI	2.32%	1.26%	1.06%
IMV	2.72%	1.42%	1.30%
Triangle	1.11%	2.60%	-1.49%

Our observations would also be justified theoretically. The AFFINE motion model can describe the complex motion models in a part of the scene that the traditional motion models such as translational motion mode, is not effective, such as the borderline CTUs. For borderline CTUs, usually, the similar corresponding block could not be found in the previous frame efficiently. For example, imagine a situation where an object that was not in the previous frames enters the scene. So, a suitable match for this object cannot be found in the previous frame using the traditional motion models such as translational mode.

The complex motion model such as zooming or rotation could be useful in such situations. The movement of the blocks in the central part of the frame is usually transitional. So, for these blocks, the traditional motion models can usually be sufficiently effective. In addition, during the image zooming, which is another main target of the AFFINE tool, in the border parts of the frame, a suitable matching will not be found using the traditional motion mode. Figure 6 shows the frames of the PartyScene sequence and the coding gain of using the AFFINE coding tool in the various parts of the frame. According to our simulation results, the AFFINE coding tool achieves much higher performance in borderline CTUs

(29.44%, on average) than the central ones where the new object enters or the frame is zooming (10.27%). So, it seems logical that the AFFINE coding tool would be more effective for borderline CTUs.

Figure 6: The consecutive frames of PartyScene sequence and the coding performance of using the AFFINE tool at central and borderline CTUs

4.1.2. Proposed approach

According to these observations, it is enough to make the AFFINE and IMV coding tools only for the borderline CTUs. This way, the computational complexity of the rate-distortion optimization process would be decreased considerably, without affecting the overall coding efficiency, significantly.

4.2. The second approach: the correlations between various coding tools

4.2.1. Observations

In the second approach, the correlation between various coding tools is used to reduce the computational complexity of the rate-distortion process. To extract the correlation, each of the six coding tools is enabled using the configuration file of VTM reference software. Then, two GoPs of all of the test sequences are encoded using four different QP values, 22, 27, 32, and 37. The bitrate and PSNR of each CTU are extracted. Then, each coding tool is disabled using the configuration file and the test sequences are encoded as before with the four QP values. Once again, the bitrate and PSNR of various CTUs are obtained. Then, the correlation coefficient between the RD performances of each two coding tools is extracted for all of the CTUs, on average.

Table 3 shows these results for the Blowing Bubbles, Four People, BasketballDrill, and BasketballPass test sequences.

Using these results, we can figure how various coding tools are correlated to each other. The high correlation coefficient between every two coding tools shows that if one of them is selected during the rate-distortion process as a proper coding tool for a CTU, because of its considerable RD performance, the other one will most likely be selected as well. Hence, only checking the RD performance of the first one in the rate-distortion process is sufficient and the other one could be selected as a proper coding tool without any RD performance evaluation. This way, the computational complexity of the rate-distortion process would be decreased considerably.

Our observations would also be justified theoretically. One of the main factors in estimating the total bitrate of a block is the selected prediction mode during the rate-distortion process. Accordingly, the coding tools that affect this choice can affect the total bitrate and the quality of the CTUs considerably and similarly. Among the examined coding tools, according to the explanations in section 2, the AFFINE and Triangle coding tools can affect the selection of the prediction modes. Therefore, the high correlation between the compression efficiency of these two coding tools seems logical.

4.2.2. Proposed approach

Using the extracted correlation coefficients for various pairs of the coding tool, we found out that the Affine and Triangle coding tools are highly correlated to each other. So, we changed the encoder part of the VTM reference software in a way that if one of these two coding tools is selected as a high-performance tool in the rate-distortion process, the other tool would be automatically selected in this process without evaluating its performance.

5. Simulation results

5.1. The first approach

To examine our proposed method, some changes are made in the encoder part of the VTM reference software. So, when the IMV or AFFINE coding modes are enabled in the configuration file, their performance improvement in the rate-distortion optimization process is estimated only for the borderline CTUs.

Then, the test sequences are encoded with four different QP values, 22, 27, 32, and 37, and the PSNR and bitrate of each CTU and the corresponding BD-rate are extracted. In this paper, the amount of time to encode the video is considered as a measure to investigate the computational complexity. The encoding time is extracted with an Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz and 8.00 GB RAM. The results are compared with the baseline approach in which these coding tools are checked for all CTUs in the rate-distortion optimization process. The encoding performance and computational complexity are shown in Table 4 and Table 5. The performance of each proposed and anchor method is estimated compared to the baseline method in which the tools are disabled for all CTUs. The results show that the amount of coding loss in our proposed approach is almost the same as the baseline method, but the computational complexity is decreased significantly.

We have also compared the results of our method with the method proposed in [24]. The proposed method in this paper states that in VVC, the AFFINE motion estimation is done through some iterations to find the best motion vectors. The changes of the MVs at the ith iteration as dMVi and the encoder try to minimize it. So, it is much faster than searching every pixel of the search area to find the best match for each block in the reference frame. The iterations of the algorithm will be stopped when the difference between the MVs of two consecutive iterations would be less than a specific threshold. Since the additional iterations cannot update the predicted MVs anymore.

The method proposed in [24] tries to minimize the number of iterations to reduce the computational complexity. In this paper, the encoding time is introduced as a metric to measure the computational complexity. So, the percentage of encoding time gain of the proposed approach over the baseline method is reported. The baseline approach in this paper is the same as the reference approach in our paper, which is the VVC test model (VTM). Simulation results show that using the method proposed in [24], the computational complexity is reduced by 2.12% for our tested sequences, which is less than the reduction in computational complexity of our proposed method which is 8.43%, with almost the same BD-rate performance.

5.2. The second Approach

Examining the results of Table 3, we concluded that Affine and Triangle coding tools are highly correlated to each other. Therefore, our second proposed solution to reduce the computational complexity of the VVC encoder suggests using this correlation as follows. During the rate-distortion optimization process, if the efficiency of using the first coding tool in encoding a CTU is high enough, it is no longer necessary to estimate the efficiency of the second coding tool using the RDO process. It can be assumed that the efficiency of the second coding tool is also high enough.

To show the performance of this approach, we have made some changes in the encoder part of the VTM reference software. These changes are such that when these two coding tools are enabled in the encoder configuration file, just the performance of one of them is estimated in the rate-distortion process. Table 6 shows the results of this approach. As we can see in this table, using this approach, the BD-rate is reduced by only 0.64%, but the encoding time gain is 4.96%, on average.

6. Conclusion

The new coding tools of the VVC standard improve the compression efficiency considerably. However, they affected the computational complexity of the rate-distortion process since the effectiveness of each coding tool should be estimated for each CTU in the rate-distortion process. In this paper, we propose two methods for rate-distortion-complexity optimization that try to examine the behavior of each coding tool at various parts of the frame and the correlation of coding tools to each other. Simulation results show that using the proposed approach, the coding efficiency of the rate-distortion process is not affected too much but the computational complexity is improved considerably.

Table 3. The correlation coefficient between every two coding tools for all CTUs on average.

Test Sequence	Correlation Coefficient						
	Tools	**Affine**	**DMVR**	**BIO**	**GBI**	**IMV**	**Triangle**
Blowing Bubbles	Affine	100%	73%	59%	47%	40%	82%
	DMVR	73%	100%	37%	67%	63%	72%
	BIO	59%	37%	100%	69%	47%	88%
	GBi	47%	67%	69%	100%	92%	79%
	IMV	40%	63%	47%	92%	100%	59%
	Triangle	82%	72%	88%	79%	59%	100%
Four People	Affine	100%	15%	59%	76%	28%	70%
	DMVR	15%	100%	37%	-21%	27%	3%
	BIO	59%	37%	100%	51%	55%	70%
	GBi	76%	-21%	51%	100%	-2%	78%
	IMV	28%	27%	55%	-2%	100%	37%
	Triangle	70%	3%	70%	78%	37%	100%
BasketballDrill	Affine	100%	59%	30%	55%	60%	57%
	DMVR	59%	100%	64%	29%	55%	37%
	BIO	30%	64%	100%	35%	43%	29%
	GBi	55%	29%	35%	100%	34%	29%
	IMV	60%	55%	43%	34%	100%	47%
	Triangle	57%	37%	29%	29%	47%	100%
BasketballPass	Affine	100%	44%	60%	82%	70%	81%
	DMVR	44%	100%	66%	20%	48%	62%
	BIO	60%	66%	100%	57%	59%	50%
	GBi	82%	20%	57%	100%	53%	65%
	IMV	70%	48%	59%	53%	100%	84%
	Triangle	81%	62%	50%	65%	84%	100%

Table 4. The encoding performance of the first proposed approach compared to the anchor methods for the AFFINE coding tool

Test Sequence	Anchor		Proposed		BD-rate loss of our proposed approach	The encoding time gain of our proposed approach over the VTM	The encoding time gain of the method proposed in [24] over the VTM
	BD-rate	Enc. Time(s)	BD-rate	Enc. Time(s)			
BasketBall Drive	3.38%	5540.95	1.86%	5042.32	-1.52%	9.00%	2.06%
BQ Terrace	0.65%	4042.14	0.57%	3823.19	-0.08%	5.42%	1.65%
Cactus	4.06%	4542.88	3.67%	4191.52	-0.39%	7.73%	2.11%
Basketball Drill	0.56%	1038.23	0.03%	977.16	-0.54%	5.88%	0.74%
BQ Mall	0.25%	815.24	0.22%	779.32	-0.04%	4.41%	1.17%
Party Scene	1.91%	1345.24	1.71%	1302.73	-0.20%	3.16%	2.53%
Race Horses big	1.29%	1823.61	0.39%	1676.62	-0.90%	8.06%	0.84%
Four People	-0.27%	786.94	-0.41%	700.30	-0.14%	11.01%	5.23%
Johnny	2.86%	766.25	-2.03%	648.36	-4.88%	15.39%	1.93%
Kristen And Sara	0.57%	842.098	-0.10%	722.40	-0.67%	14.21%	2.81%
Average					-0.94%	8.43%	2.12%

Table 5. The encoding performance of the first proposed approach compared to the anchor method for the IMV coding tool

Test Sequence	Anchor		Proposed		BD-rate loss of our proposed approach	The encoding time gain of our proposed approach over the VTM
	BD-rate	Enc. Time(s)	BD-rate	Enc. Time(s)		
BasketBall Drive	0.77%	123.51	0.31%	37.44	-0.47%	30.31%
BQ Terrace	0.98%	93.53	0.92%	26.36	-0.07%	28.18%
Cactus	0.28%	98.71	0.21%	24.69	-0.07%	25.02%
Basketball Drill	0.46%	23.41	0.01%	15.18	-0.46%	64.85%
BQ Mall	0.10%	17.93	-0.02%	10.60	-0.12%	59.15%
Party Scene	0.09%	27.94	0.05%	19.96	-0.04%	71.44%
Race Horses big	0.58%	39.72	-0.13%	27.87	-0.71%	70.15%
Four People	-0.25%	14.09	-0.48%	6.57	-0.23%	46.64%
Kristen And Sara	1.05%	15.39	0.13%	5.85	-0.92%	38.03%
Average					-0.34%	48.20%

Table 6. The encoding performance of the second proposed approach compared to the anchor method

Test sequences	Anchor method			Proposed method			BD-rate loss	The encoding time gain of our proposed approach over the VTM
	Bitrate (kbps)	PSNR	Enc. Time(s)	Bitrate (kbps)	PSNR	Enc. Time(s)		
BasketBallDrive	5382.17	38.06	7630.51	5397.42	38.05	7269.84	1.09%	3.96%
BQTerrace	1481.93	35.09	6708.34	14829.01	35.09	5636.87	0.11%	17.36%
Cactus	8495.56	36.24	6246.46	8538.75	36.23	5984.25	1.19%	2.94%
BasketballDrill	1611.72	37.57	1438.61	1614.63	37.57	1327.44	0.42%	7.66%
BQMall	2061.41	36.48	1165.01	2065.52	36.46	1085.52	0.66%	6.69%
PartyScene	48.4.33	33.65	1835.54	4819.62	33.64	1697.37	0.46%	6.68%
RaceHorses_big	2279.54	34.35	2440.89	2279.98	34.35	2261.47	0.21%	6.87%
BasketBallPass	423.13	37.42	257.83	424.62	37.41	241.88	0.82%	5.88%
BlowingBubbles	799.56	34.65	427.57	803.09	34.63	412.89	1.17%	3.14%
BQSquare	843.57	34.76	244.80	845.89	34.75	235.71	0.62%	3.35%
RaceHorses_s	647.44	34.12	573.35	648.77	34.11	556.82	0.33%	2.87%
FourPeople	1596.82	40.37	1148.83	1599.90	40.36	1106.52	0.60%	3.75%
Johnny	1195.94	41.10	1144.15	1199.60	41.10	1118.36	0.58%	1.63%
KristenAndSara	1345.08	41.30	1188.36	1352.48	41.30	1170.55	1.05%	1.45%
Balboa	31968.6	41.9	34456.6	32189.6	41.90	34118.8	0.78%	0.98%
BranCastle2	76570.4	36.04	145339	76745.4	36.04	143029	0.36%	0.39%
Broadway	35906.5	41.53	30004.4	36106.5	41.53	29040.5	0.72%	4.47%
Landing	22531.2	40.42	58116.4	22592.3	40.42	55305	0.37%	9.26%
Average	-	-	-	-	-	-	0.64%	4.96%

References

[1] M. Manohara, R. Mudumbai, J. Gibson and U. Madhow, "Error correction scheme for uncompressed HD video over wireless," *2009 IEEE International Conference on Multimedia and Expo*, pp. 802-805, 2009.

[2] M-Z. Wang, S. Wan, H. Gong, and M-Y. Ma, "Attention-Based Dual-Scale CNN In-Loop Filter for Versatile Video Coding," *IEEE Access*, vol. 7, pp. 145214 – 145226, 2019.

[3] A. Rezaeieh and H. Roodaki, "A Method for Rate-Distortion-Complexity Optimization in Versatile Video Coding Standard," *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1-5, 2021.

[4] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, and D. Menard, "Complexity Reduction Opportunities in the Future VVC Intra Encoder," *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1-6, 2019.

[5] L. Li, H. Li, D. Liu, H. Yang, S. Lin, H. Chen, and F. Wu, "An Efficient Four-Parameter Affine Motion Model for Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 1934 – 1948, 2017.

[6] X. Ji, D. Zhao, and W. Gao, "Concealment of Whole-Picture Loss in Hierarchical B-Picture Scalable Video *Coding," IEEE Transactions on Multimedia, vol. 11,* pp. 11-22, 2009.

[7] X. Chen, J. An and J. Zheng, "Decoder-Side Motion Vector Refinement Based on Bilateral Template Matching," Joint Video Exploration Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 4th Meeting: Chengdu, CN, 15–21, 2016.

[8] C-C. Chen, X. Xiu, Y. He, and Y. Ye, "Generalized Bi-prediction Method for Future Video Coding," *Picture Coding Symposium (PCS),* pp. 1-5, 2016.

[9] S. Aramvith and M-T. Sun, The Essential Guide to Video Processing (Second Edition), Academic Press, 2009.

[10] D. PARK, J. LEE, J-W. KANG, and J-G. KIM, "Simplified Triangular Partitioning Mode in Versatile Video Coding," *IEICE Transactions on Information and Systems*, vol. E103–D, pp. 472-475, 2020.

[11] W. Chien and J. Boyce, "JVET AHG report: Tool reporting procedure (AHG13)," 13th JVET Meeting, Doc. JVET-M0013, 2019.

[12] I. E. Richardson, The H.264 Advanced Video Compression Standard 2nd Edition, John Wiley and Sons, 2003.

[13] X. Li, P. Amon, A. Hutter, and A. Kaup, "Lagrange Multiplier Selection for Rate-Distortion Optimization in SVC," *Picture Coding Symposium*, pp. 1-4, 2009.

[14] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, and D. Menard, "Complexity Reduction Opportunities in the Future VVC Intra Encoder," *IEEE 21st International Workshop on Multimedia Signal Processing (MMSP),* pp. 1-6, 2019.

[15] M. Aklouf, M. Leny, F. Dufaux, and M. Kieffer, "Low Complexity Versatile Video Coding (VVC) for Low Bitrate Applications," *8th European Workshop on Visual Information Processing (EUVIP)*, pp. 22-27, 2019.

[16] H. Gao, S. Esenlik, Z. Zhao, E. Steinbach, and J. Chen, "Low Complexity Decoder Side Motion Vector Refinement for VVC," *Picture Coding Symposium (PCS),* pp. 1-5, 2019.

[17] S-H. Park and J-W. Kang, "Fast Affine Motion Estimation for Versatile Video Coding (VVC) Encoding," *IEEE Access*, vol. 7, pp. 1-1, 2019.

[18] Z. Zhang, X. Zhao, X. Li, Z. Li and S. Liu, "Fast Adaptive Multiple Transform for Versatile Video Coding," *Data Compression Conference (DCC)*, Snowbird, pp. 63-72, 2019.

[19] Q. H. Van, L. D. T. Hue, V. D. Du, V. N. Hong, and X. HoangVan, "Complexity Controlled Side Information Creation for Distributed Scalable Video Coding," *3rd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, pp. 104-108, 2019.

[20] K-H.Tai, M-J. Chen, J-R. Lin, R-Y. Huang, C.-H. Yeh, C-Y. Chen, S. D. Lin, R-M. Weng, C.-Y. Chang, "Acceleration for HEVC Encoder by Bimodal Segmentation of Rate-Distortion Cost and Accurate Determination of Early Termination and Early Split," *IEEE Access*, vol. 7, pp. 45259-45273, 2019.

[21] J. Boyce, K. Suehring, X. Li and V. Seregin, "JVET common test conditions and software reference configurations", Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting: San Diego, US, 10–20 Apr. 2018.

[22] https://jvet.hhi.fraunhofer.de/trac/vvc/browser/vtm, last access on November 2021.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," ITU T SG16/Q6, Doc. VCEG-M33, April 2001.

[24] W. Ren, W. He, and Y. Cui, "An Improved Fast Affine Motion Estimation Based on Edge Detection Algorithm for VVC," S*ymmetry* 2020, vol. 12, PP. 1143, 2020.

Amir Rezaeieh received his M.Sc. from the Faculty of Computer Engineering of K. N. Toosi University of Technology, in 2021.

Email: amir.rezaeieh@email.kntu.ac.ir

Hoda Roodaki received the B.S. and M.S. degrees in Computer engineering from the University of Tehran, and Sharif University of Technology, in 2005 and 2007 and the Ph.D. degree in Computer Architecture from the University of Tehran, Iran in 2014. Since 2015, she has been an Assistant Professor with the Computer Engineering Department, K. N. Toosi University, Tehran, Iran. Her research interests include Multi-view/3D and 360-degree video coding, Scalable video coding, video quality assessment, and cloud gaming.

Email: hroodaki@kntu.ac.ir

Paper Handling Data:

Submitted: 04-21-2021
Received in revised form: 11-13-2021
Accepted: 11-17-2021
Corresponding author: Hoda Roodaki, Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran.