Article

Context-Based Inter Mode Decision Method for Fast Affine Prediction in Versatile Video Coding

Seongwon Jung¹ and Dongsan Jun^{2,*}¹ Department of Convergence IT Engineering, Kyungnam University, Changwon 51767, Korea; swjung@kyungnam-ispl.kr² Department of Information and Communication Engineering, Kyungnam University, Changwon 51767, Korea

* Correspondence: dsjun9643@kyungnam.ac.kr

Abstract: Versatile Video Coding (VVC) is the most recent video coding standard developed by Joint Video Experts Team (JVET) that can achieve a bit-rate reduction of 50% with perceptually similar quality compared to the previous method, namely High Efficiency Video Coding (HEVC). Although VVC can support the significant coding performance, it leads to the tremendous computational complexity of VVC encoder. In particular, VVC has newly adopted an affine motion estimation (AME) method to overcome the limitations of the translational motion model at the expense of higher encoding complexity. In this paper, we proposed a context-based inter mode decision method for fast affine prediction that determines whether the AME is performed or not in the process of rate-distortion (RD) optimization for optimal CU-mode decision. Experimental results showed that the proposed method significantly reduced the encoding complexity of AME up to 33% with unnoticeable coding loss compared to the VVC Test Model (VTM).

Keywords: VVC; HEVC; affine prediction; video compression; encoding complexity; motion estimation; multi type tree

Citation: Jung, S.; Jun, D.Context-Based Inter Mode Decision Method for Fast Affine Prediction in Versatile Video Coding. *Electronics* **2021**, *10*, 1243. <https://doi.org/10.3390/electronics10111243>

Academic Editor: Jun Liu

Received: 23 April 2021

Accepted: 21 May 2021

Published: 24 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the state-of-the-art video coding standard, Versatile Video Coding (VVC) [1] was developed by Joint Video Experts Team (JVET), which was organized by ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). VVC can support significant coding performance compared to High Efficiency Video Coding (HEVC) [2]. According to [3], the VVC test model (VTM) [4] accomplished twice the coding efficiency compared to HEVC test model (HM) [5] in the random-access (RA) configuration of JVET common test conditions (CTC) [6]. In order to realize this improvement, VVC has adopted new coding tools for more elaborated inter prediction, such as affine inter prediction [7] and bi-prediction with CU-level weight (BCW) [8]. In addition, VVC also has developed Adaptive Motion Vector Resolution (AMVR) [9] and Symmetric Motion Vector Difference (SMVD) [10] as well as Advanced Motion Vector Prediction (AMVP) to save the bitrate of the motion parameters. With regard to merge estimation, VVC has adopted Geometric Partitioning Mode (GPM) [11], Merge with Motion Vector Difference (MMVD) [12], Decoder-Side Motion Vector Refinement (DMVR) [13], and Combined Inter/Intra Prediction (CIIP) [14]. Although these tools can improve the coding performance, the computational complexity of the encoder was approximately increased up to 10 times under the RA configuration compared to that of HEVC [3].

One of the main differences between HEVC and VVC is the block structure scheme. Both HEVC and VVC specify the largest coding unit as coding tree unit (CTU) with an interchangeable size based on the encoder configuration. In addition, to adapt to the various block properties, CTU could be split into four coding units (CUs) using a quadtree structure (QT). HEVC specifies the multiple partition unit types, namely coding unit

(CU), prediction unit (PU), and transform unit (TU). On the other hand, VVC substitutes them with another block structure named QT-based multi-type tree (MTT) structure so that MTT blocks can be partitioned into either binary trees (BTs) or ternary trees (TTs) to provide a variety of CU partitioning shapes. As shown in Figure 1, VVC specifies four MTT splitting types, which are vertical binary split (SPLIT_BT_VER), horizontal binary split (SPLIT_BT_HOR), vertical ternary split (SPLIT_TT_VER), and horizontal ternary split (SPLIT_TT_HOR). After a CTU is divided into four QTs (SPLIT_QT) by QT structure, each QT block can be further split into either sub-QTs or MTT blocks, while an MTT block can be only partitioned into sub-MTT blocks, as depicted in Figure 2. Note that a QT leaf node or a MTT leaf node is regarded as a CU, which is used as the basic unit of the prediction and transform processes without any further partitioning schemes. Therefore, a CU in VVC can have either a square or rectangular shape, while a CU in HEVC always has a square shape. It means that the block structure of VVC can provide better coding performance than HEVC by supporting flexible CU partitioning shapes.

Figure 1. Four MTT splitting types in VVC.

In this paper, we define the relationship between the upper CU and current CUs. The upper CU can be either a QT node with square shape or a MTT node with square or rectangular shapes covering the area of the current CU. For example, the divided QT, BT, and TT CUs can have the same upper CU, which is the QT node with $2N \times 2N$, as illustrated in Figure 2a. Similarly, a MTT node is also regarded as an upper CU for further partitioned sub-MTT nodes, as depicted in Figure 2b.

In general, inter prediction searches the motion-compensated block from the previously decoded reference frames by using conventional motion estimation (CME). CME conducts a kind of block-matching algorithm based on the translational motion model. Because CME cannot efficiently describe complex motions in natural videos, such as zooming and rotation, VVC has newly adopted not only CME but also affine motion estimation (AME) to overcome the limitations of the translational motion model at the expense of higher encoding complexity. VVC provides two affine motion models in the process of AME, which are the four-parameter affine model (*affine_4P*) and the six-parameter affine model (*affine_6P*). Because VVC performs the affine prediction in the unit of sub-block with the size of 4×4 , the affine MVs of each sub-block are derived from a two-control-point MV (CPMV₀, CPMV₁) or a three-control-point MV (CPMV₀, CPMV₁, CPMV₂) depending on *affine_4P* or *affine_6P*, respectively. As shown in Figure 3, CPMV₀, CPMV₁, and CPMV₂ indicate the motion vectors of the top-left corner control point, the top-right corner control point, and the bottom-left corner control point, respectively. In this paper, the horizontal and vertical MVs of CPMV₀, CPMV₁, and CPMV₂ are notated as (mv_{0x} , mv_{0y}), (mv_{1x} , mv_{1y}), and (mv_{2x} , mv_{2y}), respectively. For the *affine_4P*, MV (mv_x , mv_y) at center sample location (x , y) for a sub-block is derived as in Equation (1).

$$\begin{aligned} mv_x(x, y) &= \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{1y} - mv_{0y}}{W}y + mv_{0x} \\ mv_y(x, y) &= \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{1x} - mv_{0x}}{W}y + mv_{0y} \end{aligned} \quad (1)$$

Similarly, in case of the *affine_6P*, MV (mv_x, mv_y) at center sample location (x, y) for a sub-block is derived as in Equation (2):

$$\begin{aligned} mv_x(x, y) &= \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{2x} - mv_{0x}}{H}y + mv_{0x} \\ mv_y(x, y) &= \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{2y} - mv_{0y}}{H}y + mv_{0y}, \end{aligned} \quad (2)$$

where W and H indicate the width and the height of current CU, respectively. After computing affine MVs with 1/16 fractional MV accuracy, the predicted block of current CU is fetched from the reference frames in the unit of sub-block.

Figure 2. Relations between an upper CU and current CUs: (a) Example of QT node as upper CU; (b) Example of MTT node as upper CU.

Figure 3. Affine MV derivation per a sub-block: (a) Affine MV derivation for four-parameter affine model; (b) Affine MV derivation for six-parameter affine model.

While AME enables efficient prediction of the complex motions beyond the reach of translational motions, it causes heavy computational complexity in the VVC encoder. According to the tool-off test of affine prediction [15], the coding loss and the time savings are 3.11% and 20%, respectively. Figure 4a shows the distribution between affine inter mode using AME and AMVP mode using CME in the RA configuration of JVET CTC. In addition, we investigated the distribution of affine inter mode on the reference and non-reference frames, as shown in Figure 4b. In RA configuration of VVC, it specifies group-of-picture (GOP) size to support the hierarchical encoding structure along the different temporal layers (TLs). The frames with the highest TL value are treated as the non-reference frames in VVC. Figure 4b indicates that affine inter modes rarely occurred in non-reference frames. Although affine inter mode does not frequently occur during the VVC encoding process, it contributes to improving the coding efficiency. It means that the fast encoding scheme of AME should be carefully designed in order to minimize the coding loss. In this paper, we propose a fast affine prediction method to reduce the computational complexity of AME with unnoticeable coding loss.

The remainder of this paper is organized as follows. In Section 2, we review the related fast encoding schemes to reduce the computational complexity of motion estimation in HEVC or VVC. Then, the proposed method is described in Section 3. Finally, experimental results and conclusions are given in Sections 4 and 5, respectively.

Figure 4. Analysis of affine inter mode in the random-access configuration of JVET CTC: (a) Distributions between affine inter mode and AMVP mode; (b) Distributions of affine inter mode on the reference and non-reference frames.

2. Related Work

Although newly adopted VVC coding tools show the powerful compression performance compared to HEVC; it causes heavy computational complexity mainly due to the ME processes, including AME. Therefore, many researchers have studied fast encoding algorithms in the areas of ME without noticeable quality degradation. As discussed in [16,17], the increases of coding efficiency and computational complexity depend on how many reference frames are used. Therefore, reference frames should be carefully determined from the perspective of the trade-off between coding loss and complexity reduction. Pan et al. [18] reduced the number of reference frames based on the measurement with content similarity. In HEVC, a directional search pattern method [19], a rotating hexagonal pattern [20], and an adaptive search range [21] showed reasonable trade-off with the 64×64 search range to reduce the computational complexity of ME. VVC inherited several ME algorithms of HM, such as diamond search, raster search, and refinement processes [22], for fast ME process.

Another approach is the early termination of the ME process when the predefined conditions are satisfied at a certain ME process. For example, the ME process can be terminated with high probability because the recursive block-partitioning process has a strong correlation between motion parameters. In addition, the bi-directional ME process can be terminated with the PU correlations between QT structures with the different CU depth [23,24]. Shen et al. [25] proposed a fast CU split-decision algorithm and a CU depth-range estimation algorithm. It determines the early termination of CU partitioning using the neighboring CU-partitioning structure and motion amounts. In addition, Shen et al. [26] proposed a fast PU decision method using the motion activity. Tan et al. [27] addressed a three-step early CU-depth decision method by comparing the RD cost of the current CU with the average RD cost of the previously encoded CUs. Xiong et al. [28] presented a fast CU decision method using pyramid motion divergence, which exploits the correlation between the motion parameters and the CU split structure. Lee et al. [29] presented a fast CU size-selection method by comparing the average RD cost of the SKIP mode with an adaptive threshold.

Since AME for affine prediction was integrated on top of VTM 1.0 in 2018, several works have been developed to enhance the affine prediction in the middle of VVC standardization. In particular, Zhou et al. [30] proposed the concept of affine merge estimation to infer the affine motion parameters from the neighboring blocks without the explicit derivation of affine MVs. In [31], Zhang et al. proposed to omit the derivation of affine motion parameters in the case of chroma blocks with small block size. Those methods enable reduction of the total memory bandwidth and the encoding complexity compared

to the initial affine prediction on VTM 1.0. Recently, Park et al. [32] reduced the encoding complexity of AME consisting of two steps: in the first step, if the best mode of the upper CU is encoded as SKIP mode, the AME of current CU is omitted to avoid the unnecessary RD computations caused by the affine prediction. In the second step, when the best reference frame of CME is selected as the nearest reference frame in the uni-prediction of list 0 (L0), AME is only performed on the same reference frame in the prediction of L0. While this method showed marginal coding loss compared to VTM 3.0, it slightly reduced the computational complexity of AME.

3. Proposed Method

In order to design the fast affine prediction, we investigated the context correlation between upper and current CU with regard to the affine prediction. Based on the encoding information of upper CU, we checked the reasonable conditions to skip the affine inter mode of current CU. Because the posterior probability by Bayes' theorem shows the effectiveness of the conditions, we computed it with the likelihood and prior probability as $p\{(U_{cbf0} \&\& U_{!affine})|C_{affine_inter}\}$ and $p(C_{affine_inter})$ by Equation (3):

$$p\{C_{affine_inter}|(U_{cbf0} \&\& U_{!affine})\} = \frac{p\{(U_{cbf0} \&\& U_{!affine})|C_{affine_inter}\}p(C_{affine_inter})}{p(U_{cbf0} \&\& U_{!affine})}, \quad (3)$$

where C , U , “&&”, and “!” are denoted as current CU, upper CU, AND, and NOT logic operator, respectively. In addition, $cbf0$ means that the transformed non-zero coefficients do not exist in the CU. In other words, if current CU is encoded as $cbf0$, the CU could be generally regarded as a static area with stationary motions, which might not require sophisticated coding tools, such as affine prediction. Note that *affine* includes *affine_merge* as well as *affine_inter* in this paper. For example, C_{affine_inter} indicates that the best prediction mode of current CU is selected as affine inter mode in terms of rate-distortion optimization (RDO), as in Equation (4):

$$J = D + \lambda R, \quad (4)$$

where D , λ , and R represent the distortion, Lagrange multiplier, and the required bitrates in current CU, respectively.

We obtained the prior and posterior probabilities from two full-HD (FHD) test sequences encoded by VTM 10.0 under RA configuration. Note that those statistics are derived from the different three QPs to avoid the same conditions specified by JVET CTC. Table 1 shows that the current CU has low probabilities to be encoded as affine inter mode when the upper CU is not encoded as *affine* and $cbf0$. Because the distribution of the affine prediction is affected by the motion properties of the video sequences, the prior probabilities vary according to the different sequences and QP values. On the contrary, the posterior probabilities maintain similar distributions in Table 1. For example, while the prior probability of *RitualDance* sequence is increased up to 27%, the posterior probability is much smaller than the prior. Those statistics imply that a VVC encoder can avoid much of the unnecessary and redundant AME processes based on the conditions of upper CU before encoding the current CU. Based on the properties of the described contexts, the proposed method is composed of two steps: one is the early termination of affine inter mode of current CU after checking the conditions of upper CU, and the other is to determine the optimal affine model between the four-parameter and six-parameter affine model. Because an upper CU can have many current CUs, the proposed method was designed to allow for current CUs to exploit the previously encoded information derived from the QTMTT block structure. In this framework, the encoding information of either the upper or current CU are used to determine whether the AME of current CU is performed or not. If it does not satisfy with the conditions to skip the AME process, the second step can be applied for further computational complexity reduction to select the optimal affine model between *affine_4P* and *affine_6P*.

Table 1. Prior and posterior probability according to the conditions of upper CU.

Sequence	QP	Prior Probability	Posterior Probability
<i>RitualDance</i>	25	27.17%	9.01%
	30	24.84%	9.45%
	35	22.34%	9.93%
<i>BQTerrace</i>	25	13.83%	6.92%
	30	11.69%	6.11%
	35	9.52%	5.86%
Total	-	15.80%	5.54%

Figure 5 shows the block diagram of the proposed method, which modified the existing VVC encoder with the gray-shadowed early-termination rules. Firstly, if the upper CU is not encoded as affine mode and cbf_0 , it means that the current CU is likely to have the translational motion. In that case, the affine inter mode of the current CU is skipped without any AME process, as shown in Figure 5. In addition, if the current frame is the non-reference frame with the highest TL, and the best mode of the current CU is not encoded as affine merge, AME is also skipped based on the distribution of Figure 4b. Secondly, we assumed that there are correlations between the current CU and upper CU in terms of the affine model for AME. Therefore, the current CU only performs the AME using $affine_4P$ if the RD cost of $affine_4P$ is smaller than that of $affine_6P$ during the encoding of the upper CU, and the best mode of the current CU is encoded as the affine merge mode with $affine_4P$ before encoding the AME process.

Figure 5. Block diagram of the proposed method.

4. Experiment Results

The proposed method was evaluated under JVET CTC [6], as presented in Table 2, and we compared the proposed method with Park's [32] method. Our experiments were performed on top of VTM 10.0 as an anchor and were run on the experimental environments, as in Table 3.

Table 2. Test sequences and encoding parameters used in the experiments.

Sequences/ resolution/ frame rate	<i>Tango2</i>	UHD (3840 × 2160)	60 Hz
	<i>FoodMarket4</i>		60 Hz
	<i>CatRobot</i>		60 Hz
	<i>DaylightRoad2</i>		60 Hz
	<i>ParkRunning3</i>		50 Hz
	<i>MarketPlace</i>	FHD (1920 × 1080)	60 Hz
	<i>RitualDance</i>		60 Hz
	<i>Cactus</i>		50 Hz
	<i>BasketballDrive</i>		50 Hz
	<i>BQTerrace</i>		60 Hz
Encoding parameters	QP	22, 27, 32, 37	
	Num. of reference frames	2	
	Num. of encoding frames	33	
	Search range	384	
	RD optimization	Enable	
	Use hadamard	Enable	
	GOP size	16	
	Coding configuration	Random access	

Table 3. Experimental environments.

Experimental Environment	Options
Processor	Inter(R) Xeon(R) Gold 6138
RAM	256GB
Operating system	64-bit Windows 10
C++ compiler	Microsoft Visual C++ 2019

For comparison of the computational complexity, we measured the time saving (TS) of the AME encoding time (AMT), as in Equation (5):

$$TS_{AMT}(\%) = \frac{1}{4} \sum_{QP_i \in \{22, 27, 32, 37\}} \left(\frac{AMT_{org}(QP_i) - AMT_{fast}(QP_i)}{AMT_{org}(QP_i)} \times 100 \right), \quad (5)$$

where AMT_{org} and AMT_{fast} are the AMT of the anchor and the fast method to be tested, respectively. Since the time taken for AMT can be fluctuated depending on the QP value, the TS of each test sequences was represented as the average of AMT as well as the total encoding time (TET) compared with the anchor. Similar to calculating AMT, the average TS was computed from the four TET results corresponding to the different QP values. Table 4 shows the complexity reduction between the proposed method and the previous method [32]. The proposed method reduces the AMT of the VTM up to 33%, on average, compared to the anchor. The maximum and minimum TS achieve 54% and 27% in *BQTerrace* and *Cactus* sequences with unnoticeable coding loss. Compared to the previous method, the TS of the proposed method is as fast as 15% and 3% in terms of AMT and TET, on average, respectively. According to [33], there might be a difference between the percentage of computational complexity and the actual encoding time. Table 5 shows both AMT and TET on top of VTM 10.0, which represents the sum of the encoding time from QP 22, 27, 32, and 37.

Table 4. Time saving between the proposed and previous method [32] on top of VTM 10.0.

Class	Sequence Name	Park [32]		Proposed	
		TS _{AMT}	TS _{TET}	TS _{AMT}	TS _{TET}
UHD	<i>Tango2</i>	18%	1%	31%	4%
	<i>FoodMarket4</i>	14%	2%	32%	6%
	<i>CatRobot</i>	23%	5%	29%	6%
	<i>DaylightRoad2</i>	17%	1%	31%	3%
	<i>ParkRunning3</i>	12%	1%	28%	4%
	Average	17%	2%	30%	5%
FHD	<i>MarketPlace</i>	19%	4%	28%	5%
	<i>RitualDance</i>	7%	0%	38%	8%
	<i>Cactus</i>	20%	2%	27%	2%
	<i>BasketballDrive</i>	15%	1%	33%	5%
	<i>BQTerrace</i>	36%	5%	54%	3%
	Average	19%	2%	36%	5%
Overall Average		18%	2%	33%	5%

Table 5. Encoding time (second) for anchor, the previous [32], and the proposed method.

Class	Sequence Name	VTM 10.0		Park [32]		Proposed	
		AMT	TET	AMT	TET	AMT	TET
UHD	<i>Tango2</i>	22,085	161,006	18,168	159,777	15,264	155,151
	<i>FoodMarket4</i>	17,135	102,896	14,668	101,114	11,621	96,835
	<i>CatRobot</i>	22,865	134,108	17,618	127,569	16,210	125,670
	<i>DaylightRoad2</i>	30,100	188,027	24,895	185,325	20,772	181,654
	<i>ParkRunning3</i>	36,030	257,759	31,795	255,167	26,053	247,644
FHD	<i>MarketPlace</i>	7706	40,471	6255	38,985	5541	38,510
	<i>RitualDance</i>	7404	38,528	6873	38,366	4555	35,370
	<i>Cactus</i>	5144	39,340	4105	38,580	3778	38,579
	<i>BasketballDrive</i>	7043	51,514	5998	50,858	4689	48,899
	<i>BQTerrace</i>	3299	37,509	2098	35,605	1513	36,279

In order to evaluate the coding loss, we measured the Bjontegaard Delta Bit Rate (BDBR) [34]. In general, BDBR increase of 1% corresponds to BD-PSNR decrease of 0.05 dB where the positive increment of BDBR indicates the coding loss. Table 6 shows the coding loss both of the proposed method and the previous method compared to VTM 10.0. As shown in Table 6, the coding loss between the anchor and the proposed method is marginal for all test sequences. In particular, BDBR-U or BDBR-V shows better coding gain than the anchor in several sequences. For example, *DaylightRoad2* sequence showed a 1.14% coding gain in terms of BDBR-U, and *MarketPlace* sequence showed a 0.76% coding gain in BDBR-V component. It implies that the proposed method can sustainably maintain the coding loss under the deployment of our fast affine scheme. In addition, Table 7 shows the percentage of CUs that perform the early termination of AME (*affine_inter_off*) and only AME with four parameters (*affine_inter_4P*), when the proposed method was integrated on top of VTM 10.0.

Table 6. Coding loss between the proposed method and the previous method [32] on top of VTM 10.0.

Class	Sequence Name	Park [32]			Proposed		
		BDBR-Y	BDBR-U	BDBR-V	BDBR-Y	BDBR-U	BDBR-V
UHD	<i>Tango2</i>	−0.16%	0.02%	0.02%	−0.11%	−0.92%	0.12%
	<i>FoodMarket4</i>	−0.02%	0.36%	0.33%	−0.08%	0.35%	0.21%
	<i>CatRobot</i>	0.06%	0.29%	−0.11%	0.07%	−0.60%	−0.11%
	<i>DaylightRoad2</i>	0.05%	−1.55%	−0.21%	0.06%	−1.14%	−0.06%
	<i>ParkRunning3</i>	−0.02%	0.02%	0.21%	0.05%	0.03%	0.15%
	Average	−0.02%	−0.17%	0.05%	0.00%	−0.46%	0.06%
FHD	<i>MarketPlace</i>	0.00%	−0.08%	−0.31%	−0.04%	0.13%	−0.76%
	<i>RitualDance</i>	0.19%	0.41%	−0.15%	0.23%	0.18%	−0.38%
	<i>Cactus</i>	0.10%	−0.08%	0.55%	0.04%	0.54%	0.17%
	<i>BasketballDrive</i>	0.22%	0.00%	0.37%	0.16%	−1.70%	0.40%
	<i>BQTerrace</i>	0.06%	0.06%	−0.04%	0.02%	−0.37%	0.51%
	Average	0.11%	0.06%	0.08%	0.08%	−0.24%	−0.01%
Overall Average		0.05%	−0.05%	0.07%	0.04%	−0.35%	0.03%

Table 7. Distribution of affine prediction by the proposed method.

Class	Sequence Name	<i>affine_inter_off</i>	<i>affine_inter_4P</i>
UHD	<i>Tango2</i>	32%	1%
	<i>FoodMarket4</i>	30%	1%
	<i>CatRobot</i>	31%	2%
	<i>DaylightRoad2</i>	36%	5%
	<i>ParkRunning3</i>	35%	2%
	Average	33%	2%
FHD	<i>MarketPlace</i>	31%	4%
	<i>RitualDance</i>	37%	4%
	<i>Cactus</i>	30%	7%
	<i>BasketballDrive</i>	33%	4%
	<i>BQTerrace</i>	55%	1%
	Average	37%	4%
Overall Average		35%	3%

5. Conclusions

VVC has newly adopted an affine motion estimation (AME) method to overcome the limitations of the translational motion model at the expense of higher encoding complexity. In this paper, we proposed a context-based inter mode decision method for fast affine prediction that determines whether AME is performed or not in the process of rate-distortion (RD) optimization for optimal CU-mode decision. Because the fast encoding scheme of AME should be carefully designed in order to minimize the coding loss, we investigated the context correlation between upper and current CU with regard to the affine prediction. After we defined the relation between an upper CU and current CUs, we checked the reasonable conditions to skip the affine inter mode of current CU using the statistics of context correlations between upper CU and current CU. The proposed method was evaluated under JVET CTC, and we compared the proposed method with Park's [32] method on top of VTM 10.0 as an anchor. For comparison of the computational complexity, we measured the time saving (TS) of the AME time (AMT). Experimental results show that the proposed method significantly reduced the encoding complexity of AME up to 33% with unnoticeable coding loss compared to VVC Test Model (VTM). In addition, the AMT of the proposed method is as fast as 15% compared to the previous method, on average.

Author Contributions: Conceptualization, S.J. and D.J.; methodology, S.J. and D.J.; software, S.J.; formal analysis, D.J.; investigation, S.J. and D.J.; data curation, S.J.; resources, D.J.; writing—original draft preparation, S.J.; writing—review and editing, D.J.; visualization, S.J.; supervision, D.J.; project administration, D.J.; funding acquisition, D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Kyungnam University Foundation Grant, 2019.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bross, B.; Chen, J.; Liu, S.; Wang, Y. Versatile Video Coding (Draft 10). In Proceedings of the 19th Meeting Joint Video Experts Team (JVET), Teleconference (Online), 6–10 July 2020.
2. Sze, V.; Budagavi, M.; Sullivan, G.J. High efficiency video coding (HEVC). In *Integrated Circuit and Systems, Algorithms and Architectures*; Springer: Berlin, Germany, 2014; pp. 1–375.
3. Bross, F.; Li, X.; Suehring, K. AHG Report: Test Model Software Development (AHG3). In Proceedings of the 19th Meeting Joint Video Experts Team (JVET), Teleconference (Online), 6–10 July 2020.
4. Versatile Video Coding (VVC) Test Model (VTM). Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM (accessed on 21 March 2021).
5. Sullivan, G.; Ohm, J.; Han, W.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [\[CrossRef\]](#)
6. Bossen, F.; Boyce, J.; Suehring, K.; Li, X.; Seregin, V. JVET Common Test Conditions and Software Reference Configurations for SDR Video. In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018.
7. Yang, H.; Chen, H.; Zhao, Y.; Chen, J. Draft Text for Affine Motion Compensation. In Proceedings of the 11th Meeting Joint Video Experts Team (JVET), Ljubljana, Slovenia, 11–18 July 2018; Document JVET-K0565.
8. Su, Y.-C.; Chen, C.; Huang, Y.; Lei, S.; He, Y.; Luo, J.; Xiu, X.; Ye, Y. CE4-Related: Generalized Bi-Prediction Improvements Combined. In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018; Document JVET-L0197, JVET-L0296, and JVET-L0646.
9. Zhang, Y.; Han, Y.; Chen, C.; Hung, C.; Chien, W.; Karczewicz, M. CE4.3.3: Locally Adaptive Motion Vector Resolution and MVD Coding. In Proceedings of the 11th Meeting Joint Video Experts Team (JVET), Ljubljana, Slovenia, 10–18 July 2018; Document JVET-K0357.
10. Luo, J.; He, Y. CE4-Related: Simplified Symmetric MVD Based on CE4.4.3. In Proceedings of the 13th Meeting Joint Video Experts Team (JVET), Marrakech, Morocco, 9–18 January 2019; Document JVET-M0444.
11. Gao, H.; Esenlik, S.; Alshina, E.; Kotra, A.; Wang, B.; Liao, R.; Chen, J.; Ye, Y.; Luo, J.; Reuzé, K.; et al. Integrated Text for GEO. In Proceedings of the 17th Meeting Joint Video Experts Team (JVET), Brussels, Belgium, 7–17 January 2020; Document JVET-Q0806.
12. Jeong, S.; Park, M.; Piao, Y.; Park, M.; Choi, K. CE4 Ultimate Motion Vector Expression. In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018; Document JVET-L0054.
13. Sethuraman, S. CE9: Results of DMVR Related Tests CE9.2.1 and CE9.2.2. In Proceedings of the 13th Meeting Joint Video Experts Team (JVET), Marrakech, Morocco, 9–18 January 2019; Document JVET-M0147.
14. Chiang, M.; Hsu, C.; Huang, Y.; Lei, S. CE10.1.1: Multi-Hypothesis Prediction for Improving AMVP Mode, Skip or Merge Mode, and Intra Mode. In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018; Document JVET-L0100.
15. Chien, W.; Boyce, J. JVET AHG report: Tool Reporting Procedure (AHG13). In Proceedings of the 19th Meeting Joint Video Experts Team (JVET), Teleconference, 22 June–1 July 2020; Document JVET-S0013.
16. Rahaman, D.; Paul, M. Virtual view synthesis for free viewpoint video and multiview video compression using gaussian mixture modelling. *IEEE Trans. Image Process* **2018**, *27*, 1190–1201. [\[CrossRef\]](#)
17. Paul, M. Efficient multiview video coding using 3-D coding and saliency-based bit allocation. *IEEE Trans. Broadcast.* **2018**, *64*, 235–246. [\[CrossRef\]](#)
18. Pan, Z.; Jin, P.; Lei, J.; Zhang, Y.; Sun, X.; Kwong, S. Fast reference frame selection based on content similarity for low complexity HEVC encoder. *J. Vis. Commun. Represent.* **2016**, *40*, 516–524. [\[CrossRef\]](#)
19. Yang, S.-H.; Jiang, J.-Z.; Yang, H. Fast motion estimation for HEVC with directional search. *Electron. Lett.* **2014**, *50*, 673–675. [\[CrossRef\]](#)
20. Nalluri, P.; Alves, L.; Navarro, A. Complexity reduction methods for fast motion estimation in HEVC. *Signal Process. Image Commun.* **2015**, *39*, 280–292. [\[CrossRef\]](#)
21. Chien, W.; Liao, K.; Yang, J. Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 3696–3699.
22. Rosewarne, C.; Bross, B.; Naccari, M.; Sharman, K.; Sullivan, G. High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 9. In Proceedings of the 28th Meeting Joint Collaborative Team on Video Coding (JCT-VC), Torino, Italy, 13–21 July 2017; Document JCTVC-AB1002.

23. Park, S.; Lee, S.; Jang, E.; Jun, D.; Kang, J. Efficient biprediction decision scheme for fast high efficiency video coding encoding. *J. Electron. Imaging* **2016**, *25*, 063007. [[CrossRef](#)]
24. Rhee, C.; Lee, H. Early decision of prediction direction with hierarchical correlation for HEVC compression. *IEICE Trans. Inf. Syst.* **2013**, *96*, 972–975. [[CrossRef](#)]
25. Shen, L.; Zhang, Z.; An, P. Fast CU size decision and mode decision algorithm for HEVC intra coding. *IEEE Trans. Consum. Electron.* **2013**, *59*, 207–213. [[CrossRef](#)]
26. Shen, L.; Zhang, Z.; Liu, Z. Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1709–1722. [[CrossRef](#)]
27. Tan, H.; Liu, F.; Tan, Y.; Yeo, C. On fast coding tree block and mode decision for high-Efficiency Video Coding (HEVC). In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 825–828.
28. Xiong, J.; Li, H.; Wu, Q.; Meng, F. A fast HEVC inter CU selection method based on pyramid motion divergence. *IEEE Trans. Multimed.* **2014**, *16*, 559–564. [[CrossRef](#)]
29. Lee, J.; Kim, S.; Lim, K.; Lee, S. A fast CU size decision algorithm for HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 411–421.
30. Zhou, M. CE4: Test Results of CE4.1.11 on Line Buffer Reduction for Affine Mode. In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018; Document JVET-L0045.
31. Zhang, K.; Zhang, L.; Liu, H.; Wang, Y.; Zhao, P.; Hong, D. CE4: Affine Prediction with 4×4 Sub-blocks for Chroma Components (Test 4.1.16). In Proceedings of the 12th Meeting Joint Video Experts Team (JVET), Macao, China, 3–12 October 2018; Document JVET-L0265.
32. Park, S.; Kang, J. Fast Affine Motion Estimation for Versatile Video Coding (VVC) Encoding. *IEEE Access* **2019**, *7*, 158075–158084. [[CrossRef](#)]
33. Martínez-Rach, M.O.; Migallón, H.; López-Granado, O.; Galiano, V.; Malumbres, M.P. Imaging performance overview of the latest video coding proposals: HEVC, JEM and VVC. *J. Imaging* **2021**, *7*, 39. [[CrossRef](#)]
34. Bjontegaard, G. Calculation of Average PSNR Differences between RD Curves. In Proceedings of the 13th Meeting Video Coding Experts Group (VCEG), Austin, TX, USA, 2–4 April 2001; Document VCEG-M33.