

Article

Spatial Correlation-Based Motion-Vector Prediction for Video-Coding Efficiency Improvement

Xiantao Jiang ^{1,*}, Tian Song ², Takafumi Katayama ² and Jenq-Shiou Leu ³

¹ Department of Information Engineering, Shanghai Maritime University, NO.1550, Haigang Ave., Shanghai 201306, China

² Department of Electrical and Electronics Engineering, Tokushima University, 2-24, Shinkura-cho, Tokushima, Japan; tiansong@ee.tokushima-u.ac.jp (T.S.); katayama@ee.tokushima-u.ac.jp (T.K.)

³ Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, No. 43, Keelung Rd., Taipei 106, Taiwan; jsleu@mail.ntust.edu.tw

* Correspondence: xtjiang@shmtu.edu.cn

Received: 28 December 2018; Accepted: 20 January 2019; Published: 23 January 2019



Abstract: H.265/HEVC achieves an average bitrate reduction of 50% for fixed video quality compared with the H.264/AVC standard, while computation complexity is significantly increased. The purpose of this work is to improve coding efficiency for the next-generation video-coding standards. Therefore, by developing a novel spatial neighborhood subset, efficient spatial correlation-based motion vector prediction (MVP) with the coding-unit (CU) depth-prediction algorithm is proposed to improve coding efficiency. Firstly, by exploiting the reliability of neighboring candidate motion vectors (MVs), the spatial-candidate MVs are used to determine the optimized MVP for motion-data coding. Secondly, the spatial correlation-based coding-unit depth-prediction is presented to achieve a better trade-off between coding efficiency and computation complexity for interprediction. This approach can satisfy an extreme requirement of high coding efficiency with not-high requirements for real-time processing. The simulation results demonstrate that overall bitrates can be reduced, on average, by 5.35%, up to 9.89% compared with H.265/HEVC reference software in terms of the Bjontegaard Metric.

Keywords: H.265/HEVC; motion-vector prediction; video-coding efficiency; CU depth

1. Introduction

High-efficiency video coding (HEVC), also known as H.265, is the latest video-coding standard that was released in 2013 [1]. In H.265/HEVC, the maximal size of the basic coding unit (CU) is 64×64 , and the search range is a key parameter on search-quality control for motion estimation (ME). Compared with H.264/AVC, H.265/HEVC achieves about 50% bitrate saving, while computation complexity is significantly increased [2].

Before motion estimation in H.265/HEVC, motion-vector prediction (MVP) is introduced to define an accurate search center to save coding bits. The MVP is selected from a motion-vector (MV) candidate list that consist of a motion vector from neighboring units on the left of the current coding unit, a motion vector from the above neighboring units, and the motion vector of those spatially in the same position as the previously encoded frame. One of the MVs in the lists with minimum cost is selected as the final MVP. However, the fixed pattern of the MVP decision process without consideration of the reliability of the surrounding motion vectors makes it have lower estimation accuracy.

Recently, MV coding has been attracting much attention. Previous works have been divided into two categories: (1) based on spatial and temporal MVP candidates, (2) based on the higher-order motion model. The detailed descriptions of these methods are as follows:

The main ideas based on spatial and temporal MVP candidate schemes for MV coding have one assumption in common. The motion of neighboring blocks has to be similar [3–7]. A framework for a better MV and skip mode was proposed, and the predictors were selected by a rate-distortion criterion in Reference [3]. In this method, a simple spatial median is selected by using spatial and temporal redundancies in MV fields. MV coding techniques were proposed to improve coding efficiency in Reference [4], which include a priority-based derivation algorithm for spatial and temporal motion candidates, a surroundings-based candidate list, and a parallel derivation of the candidate list. This method can achieve, on average, 3.1% bitrate saving. W.H. Peng et al. introduced an interframe prediction technique that combines two MVs derived from a template and encoding block for overlapped block-motion compensation [5]. Moreover, multihypothesis prediction and motion-merge methods are used to achieve the trade-off between encoding efficiency and complexity, which achieves, on average, about 2% bitrate saving. Encoding-efficiency improvement of H.265/HEVC was proposed in Reference [6], and asymmetric motion partitioning (AMP) was used for interprediction. A new selection algorithm was proposed to improve the accuracy of prediction motion vectors in Reference [7]. Furthermore, an adaptive motion search-range algorithm was designed, and bitrate saving was only 0.16% on average. A novel MVP (NMVP) method was presented to improve coding efficiency in Reference [8], but coding complexity was higher. In a conclusion, the spatial and temporal MVP candidates lack precision, and these approaches limitedly improve the performance of MV coding, with higher coding complexity.

The main idea based on the higher-order motion model is that motion can be induced by moving objects and all kinds of camera positions and zoom changes when sequence motions are neither spatially regular nor temporally consistent [9–11]. Tok et al. describe how new motion-information coding and prediction schemes have been investigated to increase the efficiency of video coding [9,10]. Springer et al. present a scheme to perform fast, reliable, and precise rotational-motion estimation (RME) on navigation sequences [11]. However, the robustness of these methods is not high.

As a summary, coding-performance improvement is limited with low robustness in previous works. This work, an efficient MVP algorithm is proposed to further improve coding efficiency based on spatial-motion consistency correlation. Furthermore, a CU depth prediction algorithm is presented to reduce computation complexity based on spatial texture complexity correlation. Experiments confirm that the number of bits can be reduced with the proposed method. The proposed overall method can improve coding efficiency for the next-generation video-coding standards, and it is beyond H.265/HEVC.

2. Motivation for This Work

In H.265/HEVC standards, the input video is divided into a sequence of coding-tree units (CTUs), and the CTU is divided into the coding unit (CU) with a different size. The CU is a square region, each of which may be as large as 64×64 or as small as 8×8 . The prediction unit (PU) is a region defined by partitioning the CU, and PU contains MV information. Current PU sizes for intercoded CUs are $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $2N \times nD$, $nL \times 2N$, and $nR \times 2N$, where $N \in \{4, 8, 16, 32\}$.

There are three interprediction modes: InterMode, SkipMode, and MergeMode [12]. For SkipMode and InterMode, an advanced motion-vector-prediction (AMVP) technique is used to generate a motion-vector predictor among an AMVP candidate set including two spatial MVPs and one temporal MVP. For MergeMode, the Merge scheme is used to select a motion-vector predictor among a Merge candidate set containing four spatial MVPs and a temporal MVP. By using rate-distortion-optimization (RDO) processing, the encoder selects a final MVP within the candidate list for InterMode, SkipMode, or MergeMode, and transmits the index of the selected MVP to the decoder. In the case of InterMode, the sum of absolute transform differences (SATD) between the source and prediction samples is used as a distortion factor, and bits for *inter_pred_flag*, *ref_idx_1X*, *mvd_1X*, and *mvp_idx_1X* are set to coded bits. In the case of SkipMode, the prediction residual is not

transmitted for SkipMode. The coded bits include *skip_flag* and *merge_idx* that signals the position of the PU that has the best motion parameters to be used for the current PU. In the case of MergeMode, the SATD between source and prediction samples is used as a distortion factor, and bits for *merge_idx* are set to coded bits.

As shown in Figure 1, AMVP candidates of the current PU for intercoded CUs include five spatial-motion candidates: left candidate (L), bottom-left candidate (BL), top candidate (T), top-right candidate (TR), top-left candidate (TL); and two temporal candidates (C and H). Firstly, two left spatial candidates are selected, otherwise, the top spatial candidates are checked. Secondly, one temporal candidate is checked. When the selected candidate index is no more than 2, MV(0,0) candidate is added. It is noted that BL can be used when it is available, and Figure 2 shows the available BL.

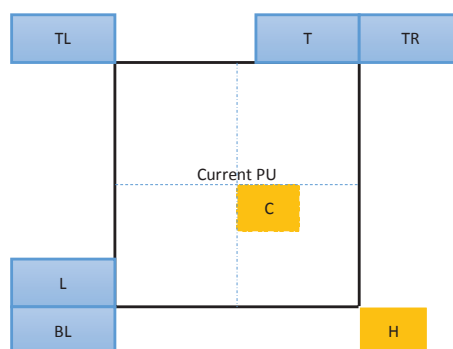


Figure 1. Advanced motion-vector-prediction (AMVP) candidates.

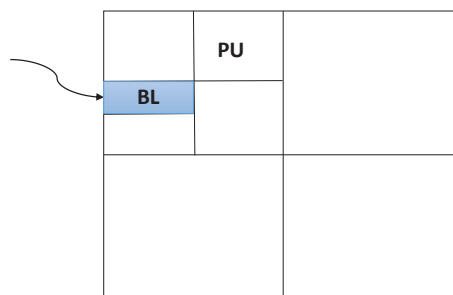


Figure 2. Bottom-left (BL) available.

Moreover, CU splitting increases the computing complexity with depth 0, 1, 2, and 3. In order to speed up the HEVC encoder, two conditions (Early_SKIP and Early_CU condition) are present, as are heuristics, to reduce the computational complexity of the HEVC. The Early_SKIP condition is that the motion-vector difference (MVD) of InterMode with $2N \times 2N$ is equal to (0, 0), and InterMode with $2N \times 2N$ contains no nonzero transform coefficients. In the Early_SKIP case, PU mode in a current CU is determined as SkipMode at the earliest possible stage. The Early_CU condition is that the best PU mode of the current CU selects the SkipMode. In the Early_CU case, the current CU is not divided into sub-CUs in the subdepth level of the current CU.

3. Proposed Method

In this section, the video-coding-efficiency improvement algorithm is described. Firstly, in order to generate a more accurate motion-vector predictor, the spatial correlations-based MVP algorithm is presented to improve encoding efficiency. Then, the spatial correlations-based CU depth-prediction algorithm is proposed to reduce computation complexity. It is noted that, in H.265/HEVC, normal mode and merge/skip mode have different methods for MVP. In the proposed approach, the method is common.

3.1. Definition of Spatial-Neighborhood Set

Considering video content with strong spatial correlations, the motion-vector predictor of the current PU for intercoded CU can be generated by the surrounding PUs. Moreover, the depth level of the current CU can be predicted from neighboring CUs where there is a similar texture or there are continuous motions.

Different from fixed-pattern AMVP technology, spatial neighborhood set G is composed of all spatial neighborhood CUs. Set G is shown in Figure 3, where CU_L , CU_{TR} and CU_{TL} denote the left, top-right, and top-left CU of the current CU, respectively. Set G is defined as

$$G = \{CU_L, \dots, CU_{TL}, \dots, CU_{TR}\} \quad (1)$$

On the one hand, the MVs and depth information of G can be used to predict the MVP and depth level of the current CU. CU contains one, two, or four PUs depending on partition mode, and each PU contains MV information. For set G , the surrounding PU directly connected with the current PU is selected in this work. Furthermore, the minimum surrounding PU size is set to 8×8 , because the MVs of the 4×8 and 8×4 surrounding PUs are not regular, and the significance of reference is small for MV prediction. On the other hand, computation complexity is high by checking all information. Therefore, a relatively reliable subset should be developed for the MVP and depth prediction. In order to utilize the spatial correlation, subset M is defined as

$$M = \{CU_L, CU_{TL}, CU_{TR}\} \quad (2)$$

where subset M is contained in set G ($M \subset G$). The basic idea of the spatial-correlation method is to prejudge the MVP and depth of the current CU according to the MVs and depths of adjacent CUs. When subset M is available, the information of M is used to predict the MVP and depth of the current CU. In contrast, when subset M is unavailable, which means that none of spatial neighborhood CUs (CU_L, CU_{TL}, CU_{TR}) exist, the information of G is used to predict the MVP and depth of the current CU.

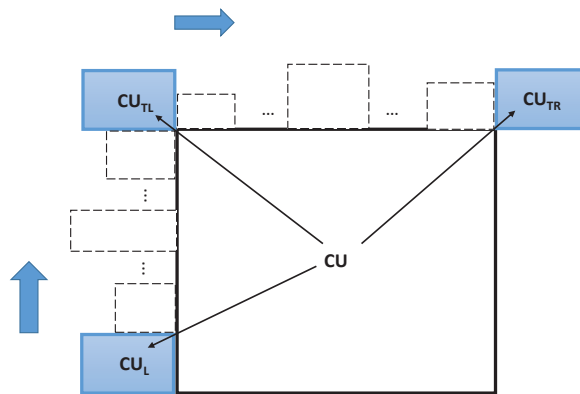


Figure 3. Spatial-correlation neighborhood set.

In this work, the spatial correlation-based method consists of two parts: a motion-vector-prediction algorithm and CU depth-prediction algorithm. Firstly, the MVP can be selected by exploiting the spatial correlation of neighboring PUs. When there is motion consistency of the neighboring PUs, a simple MV can be selected as the optimized MV for the current PU. Otherwise, more MVs of neighboring PUs can be checked to select the optimized MV. Secondly, the depth level of the current CU can be predicted by exploiting the spatial correlation of neighboring CUs. When the texture complexity of neighboring CUs tends to simple, the content of texture of the current CU tends to be not complex. On the contrary, the texture complexity of neighboring CUs tends to be complex, and the texture content of the current CU tends to be not simple.

3.2. Spatial Correlation-Based Motion-Vector-Prediction Algorithm

The performance of motion estimation highly depends on the MVP [13–16]. If the MVP is close to the calculated MV, the MVD between the MVP and the calculated MV is small, and the MVP is more accurate. However, in the H.265/HEVC standard, a total of seven spatial and temporal MVs are added to the candidate list to predict the MVP. There are two disadvantages to the current AMVP mechanism in H.265/HEVC [17]. For one thing, the number of reference MVs is limited. For another, a fixed selecting pattern is not adaptive to selecting reference MVs; therefore, it is not generating an accurate MVP by using the current AMVP mechanism. In order to further improve encoding efficiency, more reference MVs can be added to the candidate list. Owing to spatial and temporal motion consistency, the MVs surrounding the current PU are useful for determining the MVP. However, too many MVP candidates may cause a large number of calculations, so it is necessary to reduce the calculation redundancy of the MVP decision.

Based on the above-mentioned views, the reference MVs of the current PU can be used to search for an accurate MVP. In neighborhood subset M , the MVs of the neighboring CUs are shown in Figure 4, where MV_L , MV_{TR} and MV_{TL} indicate the MV candidates in the left, top right, and top left of the current PU, respectively. In H.265/HEVC, the simplified rate-distortion optimization (RDO) method is performed to estimate the motion vector [12]. In the RDO process, the rate-distortion cost (RD-cost) function ($J_{cost} = D_{distortion} + R_{bits} \times \lambda$) is minimized by the encoder, where λ is the Lagrange multiplier, $D_{distortion}$ represents the distortion between the original block and reference block, and R_{bits} represents the number of coding bits. The MVD between the MVP and the calculated MV is also signalled in the R_{bits} .

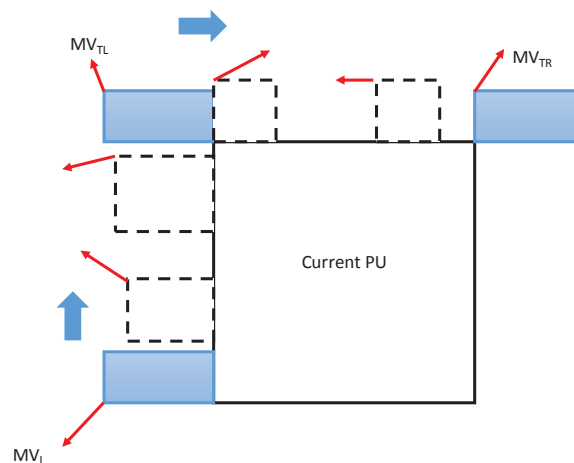


Figure 4. Reference motion vectors (MVs) of a prediction unit (PU).

For the different texture of video content, the reliability of candidate MVs can be evaluated by the MVs of spatial neighborhood subset M : $\{MV_L, MV_{TR}, MV_{TL}\}$. When these three MVs are equal, the MVs of adjacent CUs tend to the same direction. In this case, the reliability of candidate MVs is the highest, and a simple MV can be selected as the final MVP. That is, reference MVs satisfy as

$$MV_{TR} = MV_{TL} = MV_L \quad (3)$$

Then, MV_L is selected as the optimized MVP.

Furthermore, when the MV absolute difference of MV_{TL} and MV_{TR} is more than the MV absolute difference of MV_{TL} and MV_L , motion consistency in the top of PU is lower than motion consistency to the left of PU. In this case, the reliability of the MVs in the left of PU is higher than the reliability of the MVs in the top of PU. Otherwise, the reliability of the MVs in the top of PU is higher than the reliability of the MVs in the left of PU. Thus, when reference MVs satisfy as

$$|MV_{TR} - MV_{TL}| > |MV_{TL} - MV_L| \quad (4)$$

the MVP position tends to the left of PU. Then, MV_{TL} and MV_L are selected as MVP candidates. Otherwise, the MVP position tends to the top of PU, and MV_{TR} and MV_{TL} are selected as MVP candidates.

When the MVs of subset M are not available, the reliability of candidate MVs is the lowest in the spatial domain, and it is hard to obtain an accurate MVP by using the fixed AMVP mechanism. In this case, the MVP position may tend to the left of PU, and it is also possible to tend to the top of PU. Therefore, all available MVs of spatial neighborhood set G need to be checked. In order to obtain a more accurate MVP, all surrounding MVs of G can be added to the candidate MVs, and the cost of these MVs is checked to obtain an optimized MVP by comparing one with a method. When all components of G are not available, MV (0, 0) is added to the candidate list, which is the same as that in H.265/HEVC.

It is noted that the encoder codes *mvp_lx_flag* for indicating the number of coded bits for the MVP candidates. Different from the H.265/HEVC standard, in this work, the indicating method for the codec is that *mvp_lx_flag* is designed as a variable-length code, and the length of *mvp_lx_flag* is expressed as L . The relationship between the coded bits of *mvp_lx_flag* and MVP is shown as in Table 1. When M is available, the length of *mvp_lx_flag* satisfies $L = 1$ bit. However, when M is not available, the maximum value of L with a different PU size is shown in Table 2.

Table 1. Relationship between the bit of *mvp_lx_flag* and motion-vector prediction (MVP).

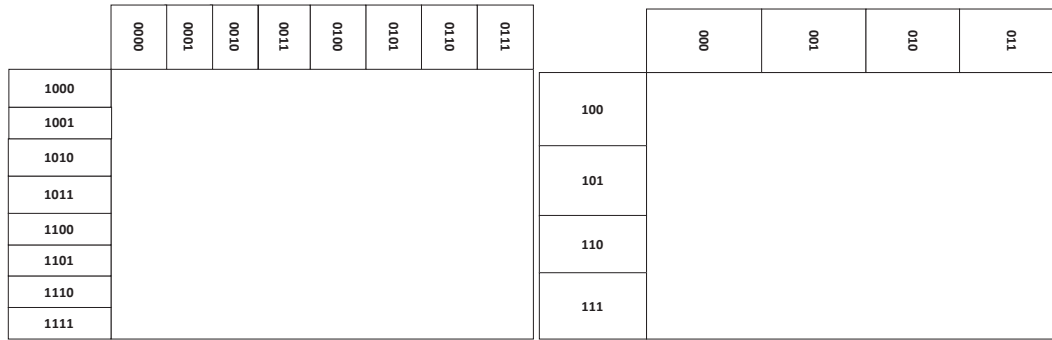
<i>mvp_lx_flag</i>	Condition	MVP
0	When subset M is selected & $ MV_{TR} - MV_{TL} > MV_{TL} - MV_L $	MV_{TL}
0	When subset M is selected & $ MV_{TR} - MV_{TL} \leq MV_{TL} - MV_L $	MV_{TR}
1	When subset M is selected & $ MV_{TR} - MV_{TL} > MV_{TL} - MV_L $	MV_L
1	When subset M is selected & $ MV_{TR} - MV_{TL} \leq MV_{TL} - MV_L $	MV_{TL}
When set G is selected		One of the selected MVP

Table 2. Length of *mvp_lx_flag* with a different PU size.

Current PU Size	Maximum Value of L
$64 \times 64, 32 \times 64, 64 \times 32, 48 \times 64, 64 \times 48, 16 \times 64, 64 \times 16$	4 bit
$32 \times 32, 16 \times 32, 32 \times 16, 24 \times 32, 32 \times 24, 8 \times 32, 32 \times 8$	3 bit
$16 \times 16, 8 \times 16, 16 \times 8, 12 \times 16, 16 \times 12, 4 \times 16, 16 \times 4$	2 bit
$8 \times 8, 4 \times 8, 8 \times 4$	1 bit

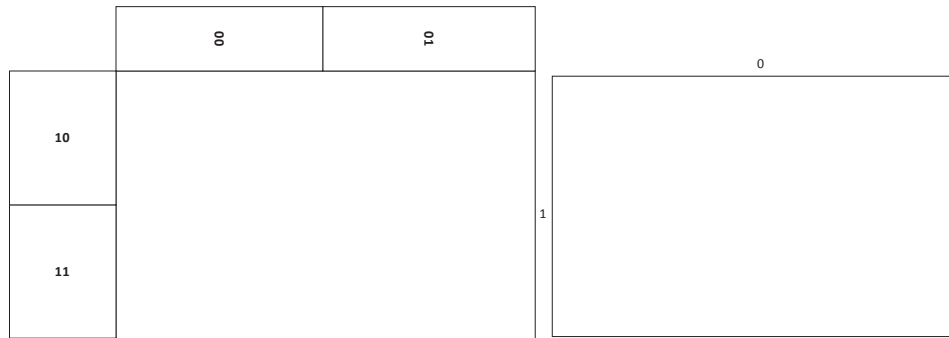
It can be seen from Table 2 that, when the current PU size is equal to 64×64 , the maximum value of *mvp_lx_flag* can be calculated as follows: (1) if the smaller surrounding PU size is 8×8 , the number of coded bits that need to index the MVP candidates is $\log_2(64/8 + 64/8) = 4$. Thus, the length of *mvp_lx_flag* satisfies $L = 4$ bit. (2) if the smaller surrounding PU size is 16×16 , the number of coded bits that need to index the MVP candidates is $\log_2(64/16 + 64/16) = 3$. Thus, the length of *mvp_lx_flag* satisfies $L = 3$ bit. (3) if the smaller surrounding PU size is 32×32 , the number of coded bits that need to index the MVP candidates is $\log_2(64/32 + 64/32) = 2$. Thus, the length of *mvp_lx_flag* satisfies $L = 2$ bit. (4) If both surrounding PUs are 64×64 in size, the number of coded bits that need to index the MVP candidates is $\log_2(64/64 + 64/64) = 1$. Thus, the length of *mvp_lx_flag* satisfies $L = 1$ bit. In this case, the maximum-value length of *mvp_lx_flag* satisfies $L = 4$ bit. Moreover, in order to clearly specify the length of *mvp_lx_flag*, Figure 5 shows the length range of *mvp_lx_flag* with 64×64 PU size. For others PU size ($32 \times 64, 64 \times 32, 48 \times 64, 64 \times 48, 16 \times 64$, and 64×16), the maximum value length of *mvp_lx_flag* satisfies $L = 4$ bit. Similarly, when the current PU size is

equal to 32×32 , the maximum value of mvp_lx_flag satisfies $L = 3$ bit. When the current PU size is equal to 16×16 , the maximum value of mvp_lx_flag satisfies $L = 2$ bit. When the current PU size is equal to 8×8 , and the smaller surrounding PU size is 8×8 , the number of coded bits that need to index the MVP candidates is $\log_2(8/8 + 8/8) = 1$. In this case, the length of mvp_lx_flag satisfies $L = 1$ bit.



(a) Smaller PU is 8×8 , $L = 4$ bit.

(b) Smaller PU is 16×16 , $L = 3$ bit.



(c) Smaller PU is 32×32 , $L = 2$ bit.

(d) PU size is 64×64 , $L = 1$ bit.

Figure 5. Length of mvp_lx_flag with a 64×64 PU size.

In H.265/HEVC standards, the distribution of the selected spatial-motion candidates is far greater than the distribution of the temporal-motion candidates [4]. In this work, the more-available spatial candidates are used to decide the MVP, and the temporal-motion candidates have little overall effect on coding efficiency. Thus, the temporal-motion candidates have been removed in the proposed method.

As per the aforementioned approaches, the spatial correlation-based motion-vector-prediction selection algorithm for interprediction is shown in Algorithm 1. Firstly, the MVP candidate list is established by using the proposed spatial-neighborhood motion vector. After that, the rate-distortion-optimal MVP is generated by executing motion estimation in the MVP candidate list, which is the search center to search for the optimal MV. Motion estimation (ME) is the process of determining a motion vector by using a block-matching algorithm [18], which is regarded as a time-consuming process. There are two advantages of this proposed method: (1) MVP accuracy was improved with the proposed method. Thus, the MVD of the current PU becomes smaller for InterMode. The length of mvp_lx_flag is variable. (2) Using the proposed method, the possibility that the MV and MVP of the current PU are consistent increased, and the probability that CU selects MergeMode increased. In the case when MVD is equal to zero, the majority of CUs select MergeMode. Therefore, by using the proposed algorithm, the effect of the proposed method (MVD becoming zero) and the effect of MergeMode overlap. The length of $merge_idx$ from the merge candidate list in MergeMode is fixed, which is the same as the definition in H.265/HEVC standards. As a result, the proposed algorithm can significantly reduce the amount of bits.

Algorithm 1: Spatial correlation-based MVP algorithm.

```

1 Start interprediction for PU
2 if subset  $M$  exist then
3   if  $MV_{TR} = MV_{TL} = MV_L$  then
4      $MV_L$  is selected as best MVP
5   else if  $|MV_{TR} - MV_{TL}| > |MV_{TL} - MV_L|$  then
6     Add  $MV_{TL}$  and  $MV_L$  to MV candidates
7   else
8     Add  $MV_{TR}$  and  $MV_{TL}$  to MV candidates
9 else
10  Add all MVs of the G to MV candidates;
11  reduce redundant MV candidates
12 Execute motion estimation to determine the rate-distortion-optimal MVP.
```

The main idea of this work is the sacrifice of computational complexity for higher coding efficiency. Thus, more MVs surrounding the current PU are added to the MV candidate list by the proposed method; therefore, most computational cost in this work is to search for an accurate MVP with the RDO process.

3.3. Spatial Correlation-Based CU Depth-Prediction Algorithm

The above spatial correlation-based MVP algorithm can significantly improve coding efficiency, while computation complexity is increased by a lot. There are quite a few related works that can reduce computation complexity [13,14,19]. However, three important issues are carefully considered to design the conditions. Firstly, arithmetic-complexity reduction is the design motivation. Secondly, the robustness of the design condition is higher. Thirdly, owing to high availability, depth information should be used. In this paper, a spatial correlation-based CU depth-prediction algorithm is presented. In order to evaluate depth-level prediction, several experiments were performed on different conditions with different configurations. In the experiments, the accuracy rate when the predicted depth level was equal to the depth level selected by the original H.265/HEVC test model was verified.

Generally, the texture complexity of image content is directly related to the depth of the image. When the depth range of the CU is higher, the texture complexity of the CU tends to be complex. On the contrary, when CU depth range is lower, CU texture complexity tends to be simple. Based on CU depth, CU texture complexity (TC) is classified into simple or complex as

$$TC = \begin{cases} \text{simple,} & \text{if } D \leq 1 \\ \text{complex,} & \text{if } D > 1 \end{cases} \quad (5)$$

where TC represents the texture complexity of a CU. D is the maximal depth of the CU in the motion-estimation processing, and default value D is set to 3 in H.265/HEVC reference software.

In this verification, the test conditions have to be carefully designed. It is clear that when the TC of the left neighboring CU_L , the top-right neighboring CU_{TR} , and the top-left neighboring CU_{TL} are simple, the TC of the current CU tends to be not complex. On the contrary, when the TC of the left neighboring CU_L , the top-right neighboring CU_{TR} , and the top-left neighboring CU_{TL} are complex, the TC of the current CU tends to be not simple. Thus, based on the above conclusions, two conditions (C1 and C2) are proposed as follows:

$$\begin{cases} \text{C1: } D_{TR} \leq 1 \&\& D_{TL} \leq 1 \&\& D_L \leq 1 \&\& D^* \leq 2 \\ \text{C2: } D_{TR} > 1 \&\& D_{TL} > 1 \&\& D_L > 1 \&\& D^* \geq 1 \end{cases}$$

where D_L , D_{TR} , D_{TL} , and D^* are the maximal depth of CU_L , CU_{TR} , CU_{TL} , and the current CU, respectively.

In order to verify the accuracy of the two conditions, accuracy rate AR is defined as

$$AR = \frac{n_1}{N} \times 100\% \quad (6)$$

while n_1 represents the number of correct-matching test cases by using the depth of the neighboring CUs to predict the depth of the current CU, and N represents the total number of test cases. In this work, four typical sequences (PeopleOnStreet, BasketballDrill, BQSquare, Vidyo1) were applied to test with low-delay (LD) and random-access (RA) profiles. From the results of Table 3, the rates of Condition 1 and Condition 2 are about 99% and 93%, respectively. That is, the depth of CU_L , CU_{TR} , and CU_{TL} has strong spatial correlation with the depth of the current CU. Thus, it is high availability to predict the depth of the current CU by utilizing the depth of the neighboring CUs.

Hence, the spatial correlation-based CU depth-prediction algorithm for interprediction is shown in Algorithm 2. Firstly, the predicted depth range of the current CU is determined by the depth of the neighboring CUs. Secondly, the RD-cost of the current CU is checked in the predicted depth range. The advantage of this method is simple and easy to achieve. Moreover, the robustness of this method is high.

Algorithm 2: Spatial correlation-based CU depth-prediction algorithm.

```

1 Start interprediction for CU
2 if (subset M exist) then
3   if ( $D_{TR} \leq 1$ ,  $D_{TL} \leq 1$  and  $D_L \leq 1$ ) then
4      $D^* \leq 2$ ;
5     Check RD-cost in predicted depth 0, 1, and 2;
6   else if ( $D_{TR} > 1$ ,  $D_{TL} > 1$  and  $D_L > 1$ ) then
7      $D^* \geq 1$ ;
8     Check RD-cost in predicted depth 1, 2, and 3;
9   else
10     $D^* \leq 3$ ;
11    Check RD-cost in predicted depth 0, 1, 2, and 3;
12 else
13   Check RD-cost in depth 0, 1, 2, and 3.
```

Table 3. Accuracy for different conditions.

Sequence	Configuration	AR	
		C1	C2
PeopleOnStreet	Low delay (LD)	96%	95%
	Random access (RA)	97%	95%
BasketballDrill	LD	99%	86%
	RA	99%	86%
BQSquare	LD	99%	95%
	RA	99%	100%
Vidyo1	LD	100%	86%
	RA	100%	100%
Average		99%	93%

3.4. Overall Algorithm

Based on the spatial-correlation model, the MVs of the neighboring PUs are used to obtain the optimized MVP. This method can improve coding efficiency, while computation complexity is increased by a lot. In order to achieve a better trade-off between coding efficiency and computation complexity, by jointing CU depth prediction, the overall algorithm can significantly improve coding performance. The flowchart of the overall algorithm is shown in Figure 6, which can be divided into three distinctive steps, as follows:

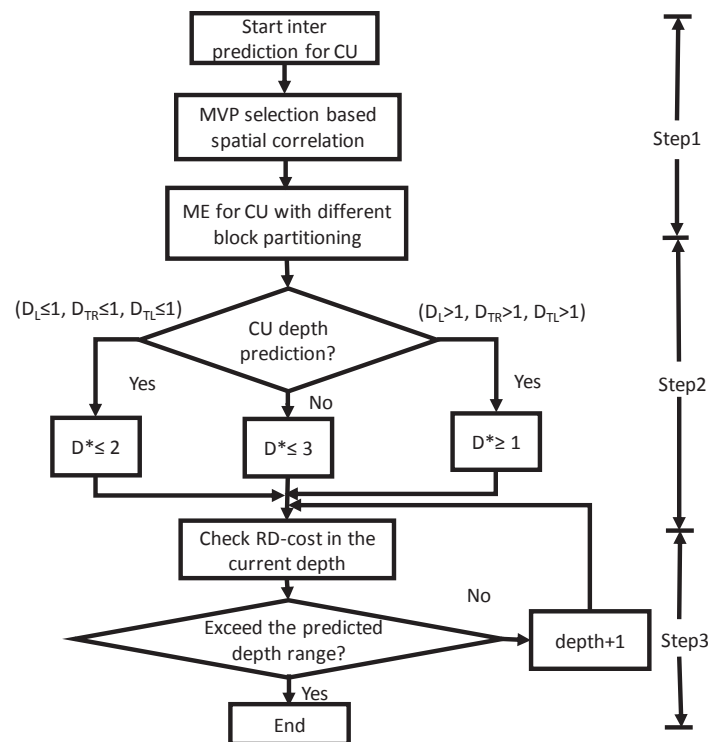


Figure 6. Flowchart of overall algorithm.

Step 1: spatial correlation-based motion-vector prediction.

The MVP is selected by using the spatial-correlation model for interprediction. Firstly, If $MV_L = MV_{TR} = MV_{TL}$, MV_L is selected as the optimized MVP. Secondly, If $|MV_{TR} - MV_{TL}| > |MV_{TL} - MV_L|$, MV_{TL} and MV_L are added to the candidates. Otherwise, MV_{TR} and MV_{TL} are added to the candidates. Lastly, If MV_{TR} , MV_{TL} , and MV_L are invalid, All MVs surrounding the current PU are added to the candidates, and the redundant MVP candidate can be reduced by comparing one with one. Executing motion estimation is to determine the rate-distortion-optimal MVP.

Step 2: spatial correlation-based CU depth prediction.

Start depth prediction with the RDO method for a CU with different block partitioning. If the maximal depths of CU_L , CU_{TR} , and CU_{TL} are less than or equal to 1, the predicted depth range of the current CU is 0, 1, and 2. Else, if the maximal depths of CU_L , CU_{TR} , and CU_{TL} are more than 1, the predicted depth range of the current CU is 1, 2, and 3. Otherwise, the predicted depth range of the current CU is 0, 1, 2, and 3.

Step 3: If the current depth of the CU exceeds the predicted depth range, RD-cost computation is stopped. Otherwise, depth is incremented by 1 and recursively checks the RD-cost in the current depth.

It should be pointed out that the overall algorithm is a recursive process, and spatial correlation-based CU depth prediction is not applied to intra.

4. Experiment Results

The proposed algorithm was implemented and verified based on H.265/HEVC reference model HM16.12 [20]. The quantization parameters (QP) were set to 22, 27, 32, and 37, respectively. The search strategy was TZsearch.

The performance of the proposed algorithm was evaluated by the Bjontegarrd delta bit rate (BDBR) and Bjontegarrd delta peak signal-to-noise rate (BDPSNR) according to the Bjontegaard metric described in Reference [21]. BDBR shows the bit-rate saving of the two methods under the same objective quality, and BDPSNR represents the difference of PSNR_Y between the two methods at the given equivalent bit rate.

There are two strategies in algorithm design, either running time for memory or memory for running time. That is to say, the more memory that can be used in a specific program to reduce running time. On the other hand, the less memory that can be used, the more running time is consumed. Thus, average complexity increased (CI) is calculated as

$$CI(\%) = \frac{1}{4} \sum_{i=1}^4 \frac{T_{pro}(QP_i) - T_{HM}(QP_i)}{T_{HM}(QP_i)} \times 100\% \quad (7)$$

where $T_{HM}(QP_i)$ and $T_{pro}(QP_i)$ are the encoding time by using the H.265/HEVC reference software and the proposed method with different QP_i .

4.1. Performance of Spatial Correlation-Based MVP Algorithm

The results of the spatial correlation-based MVP algorithm are shown in Table 4. From the experimental results, it can be seen that coding efficiency can be improved by 5.78% under the RA profile, while computation complexity is increased by 65.61%. In the aspect of coding efficiency, the method can save a 5.36% bitrate under the LD profile, while computation complexity can be increased by 61.82%. Furthermore, improvement is larger when motion activity is higher. The proposed method can save a 8.25% and 10.01% bitrate for the BasketballDrive and RaceHorses sequences, respectively. The main contribution of this work is that the proposed method can significantly improve coding efficiency for severe-motion video sequences.

Table 4. Results of the spatial correlation-based MVP algorithm.

Class	Sequence	RA			LD		
		BDBR (%)	BDPSNR (dB)	CI (%)	BDBR (%)	BDPSNR (dB)	CI (%)
1920 × 1080	Kimono	−6.30	0.225	70.89	−4.53	0.163	56.85
	ParkScene	−4.66	0.154	77.80	−4.91	0.163	58.57
	Cactus	−4.94	0.124	76.85	−4.67	0.114	63.58
	BasketballDrive	−8.25	0.179	70.89	−6.45	0.138	71.67
	BQTerrace	−2.53	0.061	79.62	−2.92	0.076	68.12
1280 × 720	Vidyo1	−4.11	0.151	68.30	−5.31	0.193	72.70
	Vidyo3	−4.77	0.167	69.60	−4.35	0.160	72.12
	Vidyo4	−5.47	0.173	69.88	−5.85	0.176	68.63
High Resolution	Average	−5.13	0.173	72.98	−4.87	0.150	66.53
832 × 480	BasketballDrill	−8.10	0.347	65.65	−7.12	0.298	55.80
	BQMall	−4.95	0.298	68.51	−5.05	0.228	62.02
	PartyScene	−4.54	0.236	60.56	−4.49	0.237	53.75
	RaceHorses	−9.51	0.408	57.28	−6.84	0.304	53.97
416 × 240	BQSquare	−2.76	0.146	64.87	−4.09	0.203	60.87
	BlowingBubbles	−5.87	0.247	58.68	−5.80	0.240	54.67
	RaceHorses	−10.01	0.532	24.72	−7.99	0.428	53.92
Low Resolution	Average	−6.53	0.306	57.18	−5.91	0.277	56.43
Average		−5.78	0.225	65.61	−5.36	0.208	61.82

The proposed algorithm is able to achieve lesser quality degradation while reducing the bitrate. The benefit is from the reliability of the candidate MVs, and the improvement in MV accuracy is significant compared with the AMVP technique. In order to evaluate this opinion, some experiments were performed to count the rate in which the MVD is equal to zero, and these rates are identified as R_x and R_y in the X-component and Y-component, respectively. Table 5 shows the results for a typical sequence (RaceHorses) when the configuration profile is RA and QP is set to 32. It is seen from the results that the MVD of most PUs is equal to zero compared with the H.265/HEVC reference software. Thus, the accuracy of the MVP was improved for InterMode, and coding efficiency was improved with the proposed method. Moreover, the rate of MVD in a whole bitstream increases with the increase of QP. Figure 7 shows the MVD portion depending on QP for the RaceHorses sequence compared to the H.265/HEVC reference model (HM). It is noted that, at a low bitrate (high QP), motion information is a major part of the total bitstream.

In H.265/HEVC standards, MergeMode is used for the PU which MVD is zero, and only the MVP index of the selected candidate in the merge list is transmitted. In other words, MergeMode allows the MV of a PU to be copied from a neighboring PU, and no motion parameter is coded in the encoder side. Correspondingly, in the decoder side, the final MV can be directly obtained by the transmitted merging MVP index. Using large block sizes for motion compensation and MergeMode is very efficient for regions with consistent displacements.

In order to analyze the percentage of the MergeMode selected as the best prediction mode, the typical sequences (RaceHorses and BasketballDrill) are tested between the proposed method and H.265/HEVC reference software, when the configure profile is RA and QP is set to 32. Figure 8 shows the percentage of MergeMode selected as the best prediction mode in the proposed method for RaceHorses and BasketballDrill sequences, compared with HEVC reference software. It noted that 76.37% and 91.46% CUs selected MergeMode as the best PU mode in the proposed method for the RaceHorses and BasketballDrill sequences, while 54.18% and 74.47% in the H.265/HEVC reference software. Therefore, more CUs select MergeMode as the best PU mode in the proposed method compared with the H.265/HEVC reference software.

To evaluate steady performance, Figure 9 shows a typical example of the R–D curve for the RaceHorses, BasketballDrive, and BQTerrace sequences in the RA and LD profiles. Regardless of in high bitrates or in low bitrates, the coding performance of the proposed method exceeded the H.265/HEVC reference model.

Table 5. Rate in which MVD is equal to zero (RaceHorses).

	R_x	R_y
H.265/HEVC reference model (HM)	85.93%	86.30%
Proposed	91.49%	91.83%

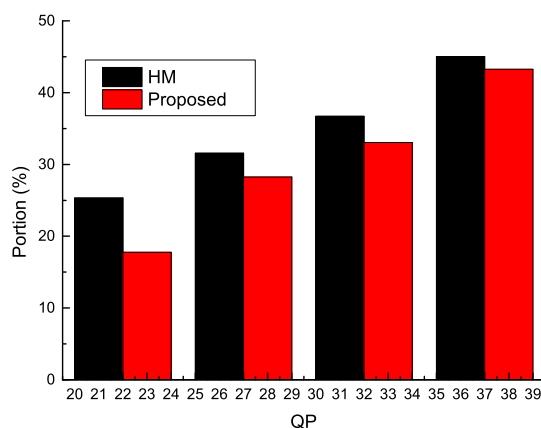
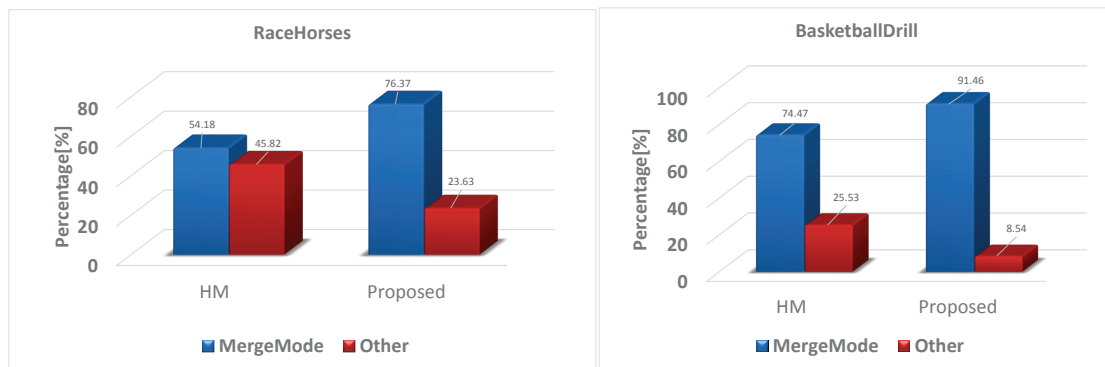
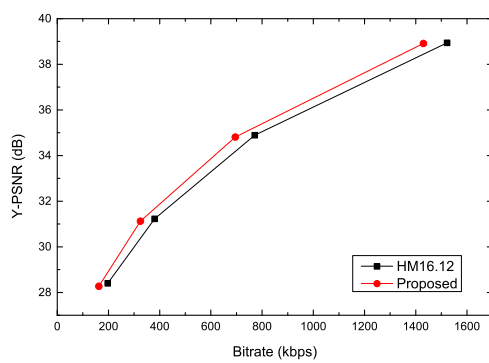


Figure 7. MVD portion depending on quantization parameters (QP).

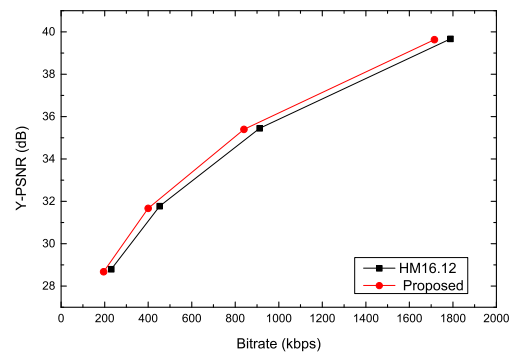


(a) RaceHorses (RA, QP = 32).

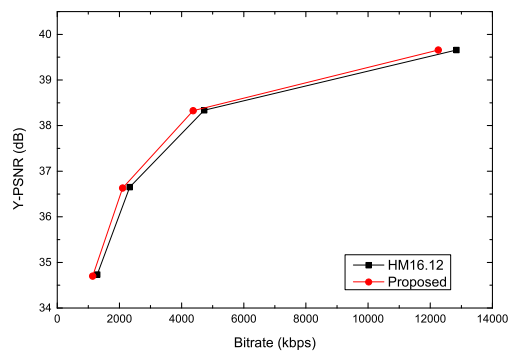
(b) BasketballDrill (RA, QP = 32).

Figure 8. Percentage of MergeMode selected as the best prediction mode (%).

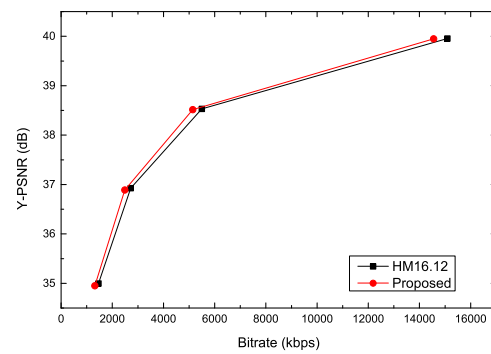
(a) RaceHorses (RA).



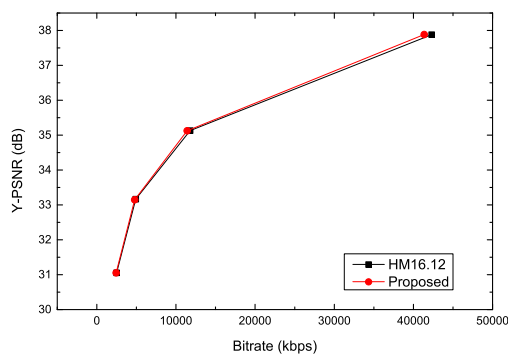
(b) RaceHorses (LD).



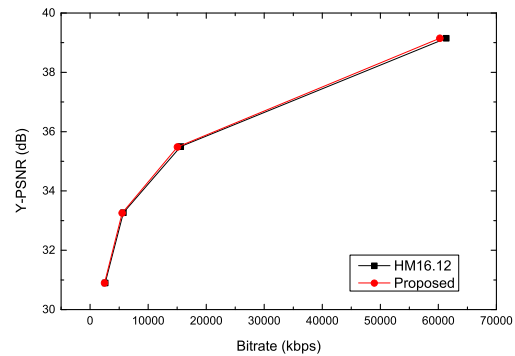
(c) BasketballDrive (RA).



(d) BasketballDrive (LD).



(e) BQTerrace (RA).



(f) BQTerrace (LD).

Figure 9. R-D curve of the spatial correlation-based MVP algorithm.

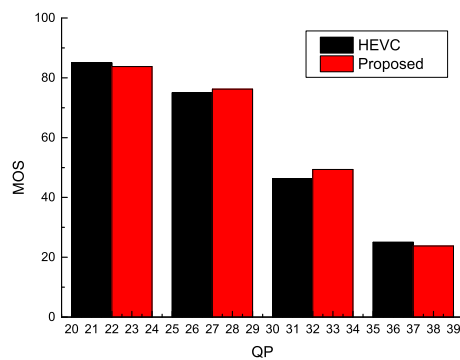
4.2. Performance of Spatial Correlation-Based CU Depth Prediction Algorithm

In order to reduce computation complexity, a spatial correlation-based CU depth-prediction algorithm is proposed. The results of this method are shown in Table 6. It can be seen that the proposed method could reduce encoding time by 12.89% under an RA profile, while coding efficiency could be reduced by 0.31%. Computation complexity could be reduced by 12.69% encoding time under the LD profile, while coding efficiency can be reduced by 0.29%. Thus, the spatial correlation-based CU depth-prediction algorithm impacts complexity reduction with a slight degradation of coding efficiency. Compared with previous complexity-reduction methods, the depth information of the neighboring CU is convenient to obtain and the implementation cost is low. Moreover, the robustness of this method is high.

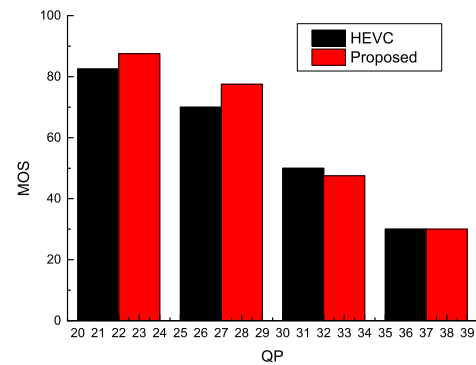
Table 6. Scale.

Scale	Mean opinion score (MOS)
Very annoying	10
Annoying	30
Slightly annoying	50
Perception but not annoying	70
Imperception	90

In order to evaluate the subjective performance for CU depth prediction, subjective tests were conducted in a controlled environment. Firstly, the Double Stimulus Impairment Scale (DSIS) method was used to perform the subjective quality-assessment experiment [22]. The subjects were presented with pairs of video sequences, where the first sequence was a H.265/HEVC reference video and the second sequence was a video with the proposed method. Secondly, a total of 24 naive viewers took part in the test campaign, the number of female and male viewers was 8 and 16, respectively, and the age median of the subjects was 25 years old. All viewers were screened for correct visual acuity and color vision. Thirdly, viewers were expected to mark their visual-quality score on an answer sheet with quality rating scale over a defined scale, and the scale was made of 5 levels ranging from “10” (Very annoying) to “90” (imperceptible) as shown in Table 7. The mean opinion score (MOS) was computed for each test as the mean across the rates of the valid subjects. For the RaceHorses and BasketballDrive sequences, Figures 10 and 11 show the MOS for male and female viewers with a different QP. Moreover, Figure 12 show the rate-MOS curves compared with the H.265/HEVC reference model. The results are reliable, and variations between the subjects were rather small. In Figure 12, it can be seen that the proposed method showed slightly improved visual quality over the H.265/HEVC reference model at higher bit rates, or H.265/HEVC reference model showed higher visual quality over the proposed method at a lower bit rate. As a whole, there was little difference in the subjective quality performance between the proposed method and the H.265/HEVC reference model.

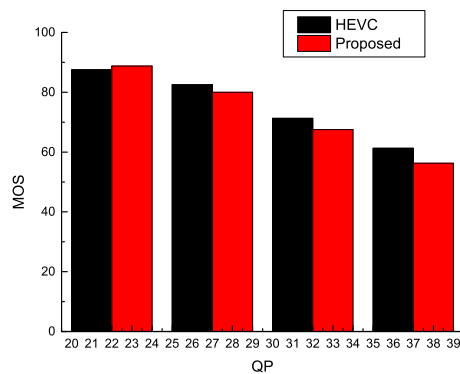


(a) Male viewers.

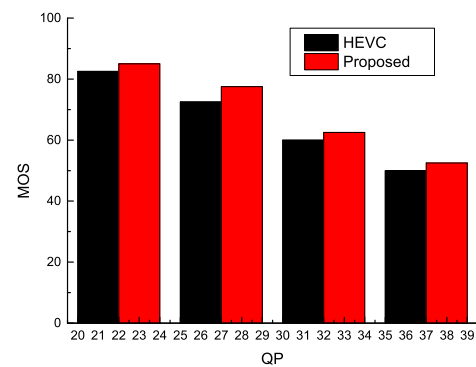


(b) Female viewers.

Figure 10. RaceHorses MOS for male and female viewers.



(a) Male viewers.



(b) Female viewers.

Figure 11. BasketballDrive MOS for male and female viewers.

Table 7. Results of the spatial correlation-based CU depth-prediction algorithm.

Class	Sequence	RA			LD		
		BDBR (%)	BDPSNR (dB)	CI(%)	BDBR (%)	BDPSNR (dB)	CI (%)
1920 × 1080	Kimono	0.00	0.000	−16.69	0.13	−0.005	−15.60
	ParkScene	0.33	−0.011	−15.71	0.20	−0.006	−12.12
	Cactus	0.44	−0.010	−14.95	0.38	−0.009	−14.83
	BasketballDrive	0.58	−0.013	−14.74	0.35	−0.007	−14.31
	BQTerrace	0.16	−0.003	−16.41	0.20	−0.005	−14.21
1280 × 720	Vidyo1	0.55	−0.015	−18.27	0.34	−0.011	−18.56
	Vidyo3	0.40	−0.012	−17.98	0.73	−0.025	−19.85
	Vidyo4	0.30	−0.009	−20.34	0.45	−0.012	−22.42
High Resolution	Average	0.34	−0.010	−16.89	0.35	−0.010	−16.49
832 × 480	BasketballDrill	0.43	−0.018	−8.50	0.42	−0.018	−10.79
	BQMall	0.61	−0.028	−9.74	0.57	−0.025	−10.80
	PartyScene	0.14	−0.007	−6.96	0.02	−0.001	−9.51
	RaceHorses	0.36	−0.015	−3.92	0.34	−0.015	−8.82
416 × 240	BQSquare	0.01	−0.001	−4.85	0.05	−0.003	−10.13
	BlowingBubbles	0.24	−0.001	−4.63	0.10	−0.004	−7.06
	RaceHorses	0.09	−0.004	−19.64	0.14	−0.008	−1.38
Low Resolution	Average	0.27	−0.010	−8.32	0.23	−0.010	−8.36
Average		0.31	−0.010	−12.89	0.29	−0.010	−12.69

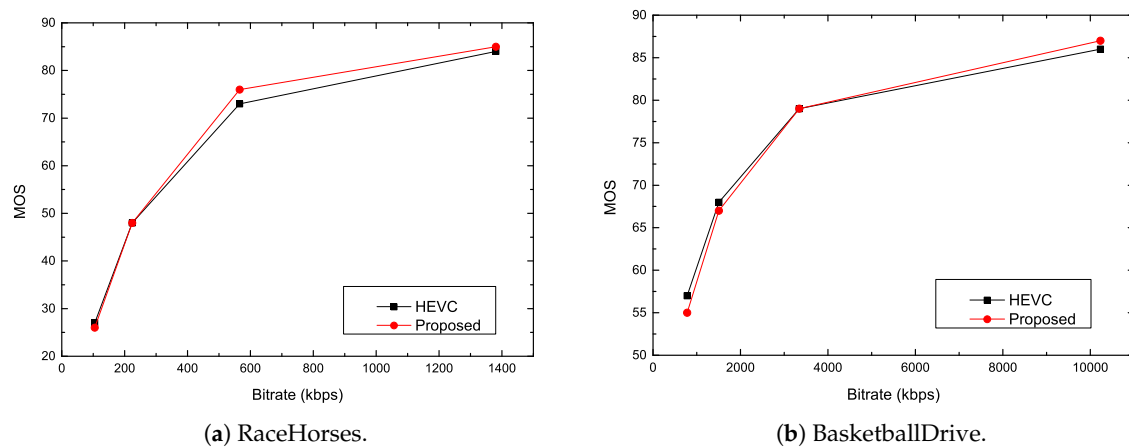


Figure 12. Rate–MOS curves.

4.3. Performance of Overall Algorithm

In order to achieve a better trade-off between coding efficiency and computation complexity, Table 8 shows the performance of the overall algorithm that jointed the spatial correlation-based MVP-prediction and CU depth-prediction algorithms. The third and fifth columns in the table show performance under the RA profile. From the experimental results, it can be seen that the coding efficiency of this algorithm could be improved by 5.35%, while computation complexity was increased by only 40.30%. The sixth and eighth columns in the table show performance under the LD profile. In the aspect of coding efficiency, the method could save 4.98% bitrate, while computation complexity could be increased by only 40.75%. Compared with the aforementioned MVP-prediction algorithm, the joint algorithm could significantly improve coding efficiency.

Table 8. Results of overall algorithm.

Class	Sequence	RA			LD		
		BDBR (%)	BDPSNR (dB)	CI (%)	BDBR (%)	BDPSNR (dB)	CI (%)
1920 × 1080	Kimono	−6.10	0.217	33.31	−4.33	0.156	33.21
	ParkScene	−4.18	0.138	34.65	−4.62	0.152	40.83
	Cactus	−4.40	0.110	35.11	−4.15	0.102	36.04
	BasketballDrive	−7.45	0.161	36.60	−5.81	0.124	39.57
	BQTerrace	−2.04	0.050	36.56	−2.70	0.070	38.85
1280 × 720	Vidyo1	−3.48	0.129	38.68	−4.53	0.162	41.39
	Vidyo3	−4.39	0.153	37.44	−4.14	0.142	36.86
	Vidyo4	−4.85	0.151	35.04	−5.24	0.158	34.82
High Resolution	Average	−4.61	0.140	35.92	−4.44	0.130	38.07
832 × 480	BasketballDrill	−7.44	0.319	48.57	−6.45	0.269	45.97
	BQMall	−3.94	0.181	49.19	−3.83	0.173	47.04
	PartyScene	−4.40	0.229	45.81	−4.40	0.232	38.47
	RaceHorses	−9.27	0.398	47.28	−6.86	0.306	41.65
416 × 240	BQSquare	−2.80	0.148	52.49	−4.07	0.203	41.03
	BlowingBubbles	−5.67	0.238	47.94	−5.71	0.237	42.33
	RaceHorses	−9.89	0.527	25.89	−7.92	0.426	50.26
Low Resolution	Average	−6.20	0.291	45.31	−5.61	0.264	43.82
Average		−5.35	0.210	40.30	−4.98	0.194	40.75

To evaluate steady performance, Figure 13 shows a typical example of the R–D curve for the RaceHorses, BasketballDrive, and BQTerrace sequences in the RA and LD profiles. Regardless of in

high bitrates or in low bitrates, the coding efficiency of the overall algorithm significantly improved coding performance.

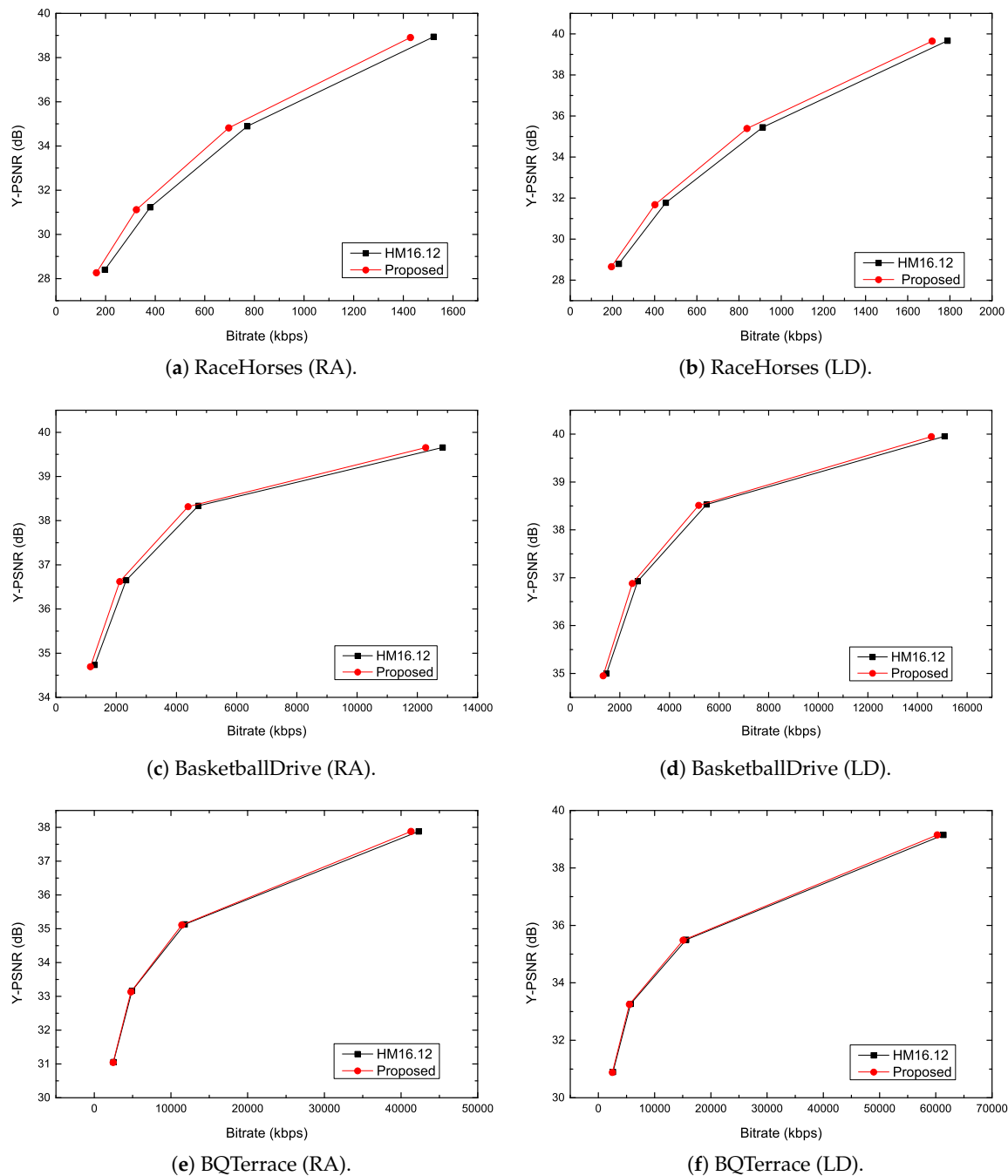


Figure 13. R-D curve of the overall algorithm.

Compared with previous work [4,5], the reference results are shown in Table 9. In both high resolution and low resolution, the coding efficiency of the proposed method was higher than Lin's and Peng's method. The benefit is from an accurate MVP according to the MVs surrounding the CU. Moreover, this proposed method can achieve a better trade-off between coding efficiency and computation complexity.

It should be specially mentioned that this proposed method causes coding complexity to increase by raising coding efficiency. However, for the application that does not care about real-time encoding, and cares more about coding efficiency, it is an efficient approach for coding-efficiency improvement.

Moreover, the redundancy computational could be further reduced. In the future, this increased coding complexity could be further reduced by parallel computation.

Table 9. Performance comparison with previous work.

Configuration	Method	BDBR (%)		
		High Res.	Low Res.	Average
RA	Proposed	−4.61	−6.20	−5.35
	J.L Lin [4]	−2.30	−2.20	−2.23
	W.H. Peng [5]	−1.70	−1.80	−1.77
LD	Proposed	−4.44	−5.61	−4.98
	J.L Lin [4]	−3.90	−4.25	−4.20
	W.H. Peng [5]	−1.85	−2.15	−2.00

It is worth noting that the purpose of this work was to improve coding efficiency for the next-generation video-coding standards, and the proposed MVP algorithm is beyond a standard HEVC structure. When using the proposed algorithm on the decoder side, additional MVP generation that is a similar process as in the encoder side can be fixed. However, this process does not need to calculate RD-cost again in decoder side. The modified decoder ensures correct decoding, although it induces a slight computation-complexity increase in the decoder side.

5. Conclusions

In this work, a spatial correlation-based motion-vector-prediction method is presented to improve coding efficiency for future video coding. Firstly, a spatial neighborhood set was introduced to describe the strong correlation between current PU and neighboring PUs. Secondly, based on spatial-motion consistency correlation, an efficient MVP algorithm was presented to improve coding performance. Furthermore, based on spatial-texture complexity correlation, a CU depth-prediction algorithm was proposed to achieve a better trade-off between coding efficiency and computation complexity. Finally, simulation results demonstrate that the proposed overall algorithm could improve coding efficiency by 4.98%–5.35% on average.

Author Contributions: X.J. designed the algorithm, conducted all experiments, analyzed the results, and wrote the manuscript. T.S. conceived the algorithm and wrote the manuscript. T.K. conducted the literature review and wrote the manuscript. J.-S.L. wrote the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number 61701297, China Postdoctoral Science Foundation grant number 2018M641982, and JSPS KAKENHI grant number 17K00157.

Acknowledgments: This research was sponsored by the National Natural Science Foundation of China (NSFC, NO.61701297), and China Postdoctoral Science Foundation (CPSF, NO. 2018M641982). It was also supported by JSPS KAKENHI, Grant Number 17K00157.

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

References

1. Sullivan, G.J.; Ohm, J.R.; Han, J.R.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668.
2. Jiang, X.; Song, T.; Shi, W.; Shimamoto, T.; Wang, L. Fast coding unit size decision based on probabilistic graphical model in high efficiency video coding inter prediction. *IEICE Trans. Inf. Syst.* **2016**, *99*, 2836–2839.
3. Laroche, G.; Jung, J.; Pesquet-Popescu, B. RD optimized coding for motion vector predictor selection. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1247–1257.
4. Lin, J.L.; Chen, Y.W.; Huang, Y.W.; Lei, S.M. Motion vector coding in the HEVC standard. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 957–968.

5. Peng, W.H.; Chen, C.C. An interframe prediction technique combining template matching prediction and block-motion compensation for high-efficiency video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1432–1446.
6. Kim, I.K.; Lee, S.; Cheon, M.S.; Lee, T.; Park, J. Coding efficiency improvement of HEVC using asymmetric motion partitioning. In Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Seoul, South Korea, 27–29 June 2012; pp. 1–4.
7. Chien, W.D.; Liao, K.Y.; Yang, J.F. Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 3696–3699.
8. Zhang, Y.; Wang, H.; Li, Z. Fast coding unit depth decision algorithm for interframe coding in HEVC. In Proceedings of the Data Compression Conference (DCC), Snowbird, UT, USA, 20–22 March 2013; pp. 53–62.
9. Tok, M.; Glantz, A.; Krutz, A.; Sikora, T. Parametric motion vector prediction for hybrid video coding. In Proceedings of the Picture Coding Symposium (PCS), Krakow, Poland, 7–9 May 2012; pp. 381–384.
10. Tok, M.; Eiselein, V.; Sikora, T. Motion modeling for motion vector coding in HEVC. In Proceedings of the Picture Coding Symposium (PCS), Cairns, QLD, Australia, 31 May–3 June 2015; pp. 154–158.
11. Springer, D.; Simmet, F.; Niederkorn, D.; Kaup, A. Robust Rotational Motion Estimation for efficient HEVC compression of 2D and 3D navigation video sequences. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 1379–1383.
12. Rosewarne, C. High Efficiency Video Coding (HEVC) Test Model 16 (HM 16); Document JCTVC-V1002, JCT-VC; October 2015. [[CrossRef](#)]
13. Lei, J.; Duan, J.; Wu, F.; Ling, N.; Hou, C. Fast Mode Decision Based on Grayscale Similarity and Inter-View Correlation for Depth Map Coding in 3D-HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 706–718.
14. Chen, M.; Wu, Y.; Yeh, C.; Lin, K.; Lin, S.D. Efficient CU and PU Decision Based on Motion Information for Interprediction of HEVC. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4735–4745.
15. Jiang, X.; Song, T.; Shimamoto, T.; Shi, W.; Wang, L. Spatio-Temporal Prediction Based Algorithm for Parallel Improvement of HEVC. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2015**, *98*, 2229–2237.
16. Kazui, K.; Shimada, S.; Koyama, J.; Nakagawa, A. Improvement on Simplified Motion Vector Prediction; Document JCTVC-E062, JCT-VC; March 2011. [[CrossRef](#)]
17. Park, J.; Park, S.; Jeon, B. Improvement on Median Motion Vector of AMVP; Document JCTVC-D095, JCT-VC; January 2011. [[CrossRef](#)]
18. Shilpa, M.; Sanjay, T. *Motion Estimation Techniques for Digital Video Coding*; Springerbriefs in Applied Sciences and Technology; Springer: Delhi, India, 2013. [[CrossRef](#)]
19. Zhou, C.; Zhou, F.; Chen, Y. Spatio-temporal correlation-based fast coding unit depth decision for high efficiency video coding. *J. Electron. Imaging* **2013**, *22*, 043001.
20. Bossen, F. Common Test Conditions and Software Reference Configurations; JCTVC-L1100, JCT-VC; January 2013. [[CrossRef](#)]
21. Bjontegaard, G. Calculation of Average PSNR Differences between RD-Curves; ITU-T SG16 Q.6, VCEG-M33, JVT; April 2001. [[CrossRef](#)]
22. Hanhart, P.; Rerabek, M.; Simone, F.D.; Ebrahimi, T. Subjective quality evaluation of the upcoming HEVC video compression standard. *Proc. SPIE* **2012**, *8499*, 84990V.

