electronics

MDPI

Article

Design of Efficient Perspective Affine Motion Estimation/Compensation for Versatile Video Coding (VVC) Standard

Young-Ju Choi [1], Dong-San Jun [2], Won-Sik Cheong [3] and Byung-Gyu Kim [1,*]

[1] Department of IT Engineering, Sookmyung Women's University, Seoul 04310, Korea; yj.choi@ivpl.sookmyung.ac.kr
[2] Department of Information and Communication Engineering, Kyungnam University, Changwon 51767, Korea; dsjun9643@kyungnam.ac.kr
[3] Immersive Media Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Korea; wscheong@etri.re.kr
* Correspondence: bg.kim@sookmyung.ac.kr; Tel.: +82-2-2077-7293

Received: 14 August 2019; Accepted: 2 September 2019; Published: 5 September 2019



Abstract: The fundamental motion model of the conventional block-based motion compensation in High Efficiency Video Coding (HEVC) is a translational motion model. However, in the real world, the motion of an object exists in the form of combining many kinds of motions. In Versatile Video Coding (VVC), a block-based 4-parameter and 6-parameter affine motion compensation (AMC) is being applied. In natural videos, in the majority of cases, a rigid object moves without any regularity rather than maintains the shape or transform with a certain rate. For this reason, the AMC still has a limit to compute complex motions. Therefore, more flexible motion model is desired for new video coding tool. In this paper, we design a perspective affine motion compensation (PAMC) method which can cope with more complex motions such as shear and shape distortion. The proposed PAMC utilizes perspective and affine motion model. The perspective motion model-based method uses four control point motion vectors (CPMVs) to give degree of freedom to all four corner vertices. Besides, the proposed algorithm is integrated into the AMC structure so that the existing affine mode and the proposed perspective mode can be executed adaptively. Because the block with the perspective motion model is a rectangle without specific feature, the proposed PAMC shows effective encoding performance for the test sequence containing irregular object distortions or dynamic rapid motions in particular. Our proposed algorithm is implemented on VTM 2.0. The experimental results show that the BD-rate reduction of the proposed technique can be achieved up to 0.45% and 0.30% on Y component for random access (RA) and low delay P (LDP) configurations, respectively.

Keywords: video coding; motion estimation; motion compensation; affine motion model; perspective motion model; VVC

1. Introduction

Video compression standard technologies are increasingly becoming more efficient and complex. With continuous development of display resolution and type along with enormous demand for high quality video contents, video coding also plays a key role in display and content industries. After standardizing H.264/AVC [1] and H.265/HEVC [2] successfully, Versatile Video Coding (VVC) [3] is being standardized by the Joint Video Exploration Team (JVET) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). Obviously, the HEVC is a reliable video compression standard. Nevertheless, more efficient video coding scheme is required for higher-resolution and the newest services such as UHD and VR.

To develop the video compression technologies beyond HEVC, experts in JVET have been actively conducting much research. VVC provides a reference software model called as VVC Test Model (VTM) [4]. At the 11th JVET meeting, VTM2 [5] was established with the inclusion of a group of new coding features as well as some of HEVC coding elements.

The basic framework of VVC is the same as HEVC, which consists of block partitioning, intra and inter prediction, transform, loop filter and entropy coding. Inter prediction, which aims to obtain a similar block in the reference frames in order to reduce the temporal redundancy, is an essential part in video coding. The main tools for inter prediction are motion estimation (ME) and motion compensation (MC). Finding precise correlation between consecutive frames is important to final coding performance. Block matching based ME and MC have been implemented in the reference software model of the previous video compression standards such as H.264/AVC and H.265/HEVC. The fundamental motion model of the conventional block-based MC is a translational motion model. In the early research, a translational motion model-based MC cannot address complex motions in natural videos such as rotation and zooming. Such being the case, during the development of the video coding standards, further elaborate models are required to handle non-translational motions.

Non-translational motion model-based studies have also been presented in the early research on video coding. Seferidis [6] and Lee [7] proposed deformable block based ME algorithms, in which all motion vectors (MVs) at any position inside a block can be calculated by using control points (CPs). Besides, Cheung and Siu [8] proposed to use the neighboring block's MVs to estimate the affine motion transformation parameters and added an affine mode. After those, affine motion compensation (AMC) has begun to attract attention. A local zoom motion estimation method was proposed to achieve more coding gain by Kim et al. [9]. In this method, they used to estimate some zoom-in/out cases of the object or background part. However they dealt with just zoom motion cases using the H.264/AVC standard.

Later, Narroschke and Swoboda [10] proposed an adjusted AMC to HEVC coding structure by investigates the use of an affine motion model with analyzing variable block size. Huang et al. [11] extended the work in [8] for HEVC and included the affine skip/direct mode to improve coding efficiency. Also, Heithausen and Vorwerk [12] investigated different kinds of higher order motion models. Moreover, Chen et al. [13] proposed the affine skip and merge mode. In addition, Heithausen [14] developed a block-to-block translational shift compensation (BBTSC) technique which related to the advanced motion vector prediction (AMVP) [15] and improved the BBTSC algorithm by applying the translational motion vector field (TMVF) in [16]. Li [17] proposed the six-parameter affine motion model and extended by simplifying model to four-parameter and adding gradient-based fast affine ME algorithm in [18]. Because the trade-off between the complexity and coding performance is attractive, the scheme in [18] was proposed to JVET [19] and was accepted as one of the core modules of Joint Exploration Model (JEM) [20,21]. After that, Zhang [22] proposed a multi model AMC approach. At the 11th JVET meeting in July 2018, modified AMC of JEM was integrated into VVC and Test Model 2 (VTM2) [5] based on [22].

Although AMC has significantly improved performance over the conventional translational MC, there is still a limit to finding complex motion accurately. Affine transformation is a model that maintains parallelism based on the 2D plane, and thus cannot work efficiently for some sequences containing object distortions. In actual videos, motion by a non-affine transformation appears more generally than by an affine transformation with such restriction. Figure 1 shows an example of a non-affine transformation in nature video. When a part of an object is represented by a rectangle, the four vertices must operate independently of each other to illustrate the deformation of the object most similarly. Even though different frames have the same object, if the depth or viewpoint information changes, the motion can not be completely estimated by affine transformation model. For this reason, more flexible motion model is desired for new coding tool to raise the encoding quality.

The method using basic warping transformation model results in high computational complexity and bit overhead because of the large number of parameters. Therefore, it is necessary to apply a model

that is not greatly increased for bit overhead compared to the existing AMC and has flexibility enough to replace the warping transformation model.

In this paper, we propose a perspective affine motion compensation (PAMC) method which improve coding efficiency compared with the AMC method of VVC. Compared to prior-arts, this paper presents two practical contributions to AMC. First, a perspective transformation model is designed in the form of MVs so that it can be used in AMC. It is an eight parameter based motion model that requires four CPMVs. Second, we propose a multi-motion model switch approach based framework to operate adaptively with AMC. In other words, six and four parameter model-based AMC and eight parameter-based perspective ME/MC are performed to select the best coding mode adaptively.

This paper is organized as follows. In Section 2, we first present AMC in VVC briefly. The proposed perspective affine motion compensation (PAMC) is introduced in Section 3. The experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

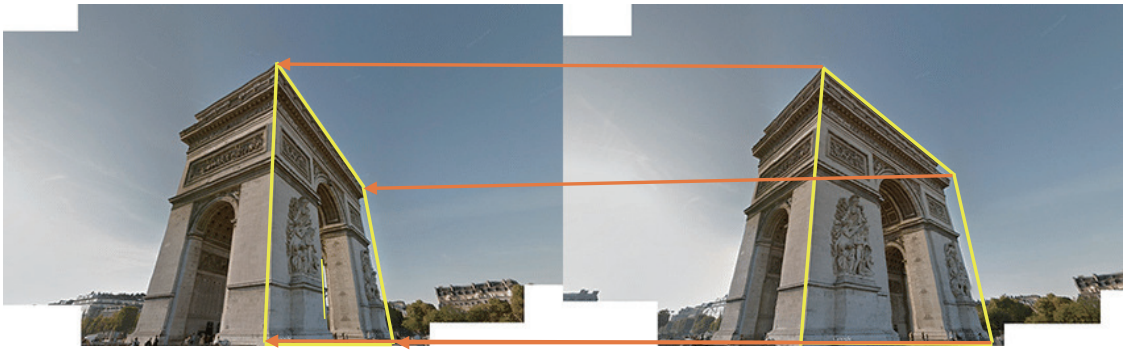


Figure 1. Example of a non-affine transformation [23].

2. Affine Motion Estimation/Compensation in VVC

HEVC standard apply translational motion model to find a corresponding prediction block. The translational motion model cannot describe complex motion such as rotation and zooming. Moreover, it cannot represent combined multiple motion. In VVC, an affine motion compensation (AMC) is implemented which supports 4-parameter and 6-parameter motion model. The motion model for the AMC prediction method in the VVC is defined for three motions: translation, rotation and zooming. Affine transformation is based on the use of a 6-parameter model. Furthermore, a simplified 4-parameter model is applied for AMC in VVC. In addition, two affine motion modes namely affine inter-mode and affine merge-mode are added to AMC module. If affine inter-mode is used for a coding unit (CU), algorithm for affine inter-mode is designed to predict the MVs at CPs. In prediction process, a gradient-based ME algorithm is used as an encoder. When a CU is applied in affine merge-mode, the MVs at CPs are derived from the spatial neighbouring CU.

2.1. 4-Parameter and 6-Parameter Affine Model

As shown in Figure 2, the affine motion vector field (MVF) of a CU is described by control point motion vectors (CPMVs): (a) two CPs (4-parameter) or (b) three CPs (6-parameter). CP_0, CP_1 and CP_2 are defined as the top-left, top-right and bottom-left corners. For 4-parameter affine motion model, MV at sample position (x, y) in a CU is derived as

$$\begin{cases} mv^h(x,y) = \frac{mv_1^h - mv_0^h}{W}x - \frac{mv_1^v - mv_0^v}{W}y + mv_0^h, \\ mv^v(x,y) = \frac{mv_1^v - mv_0^v}{W}x + \frac{mv_1^h - mv_0^h}{W}y + mv_0^v. \end{cases} \quad (1)$$

For 6-parameter affine motion model, MV at sample position (x, y) in a CU is derived as

$$\begin{cases} mv^h(x,y) = \frac{mv_1^h - mv_0^h}{W}x + \frac{mv_2^h - mv_0^h}{H}y + mv_0^h, \\ mv^v(x,y) = \frac{mv_1^v - mv_0^v}{W}x + \frac{mv_2^v - mv_0^v}{H}y + mv_0^v. \end{cases} \quad (2)$$

where (mv_0^h, mv_0^v), (mv_1^h, mv_1^v) and (mv_2^h, mv_2^v) are MVs of CP_0, CP_1 and CP_2 respectively. W and H present width and height of the current CU. The $mv^h(x,y)$ and $mv^v(x,y)$ are the horizontal and vertical components of MV for the position (x,y).

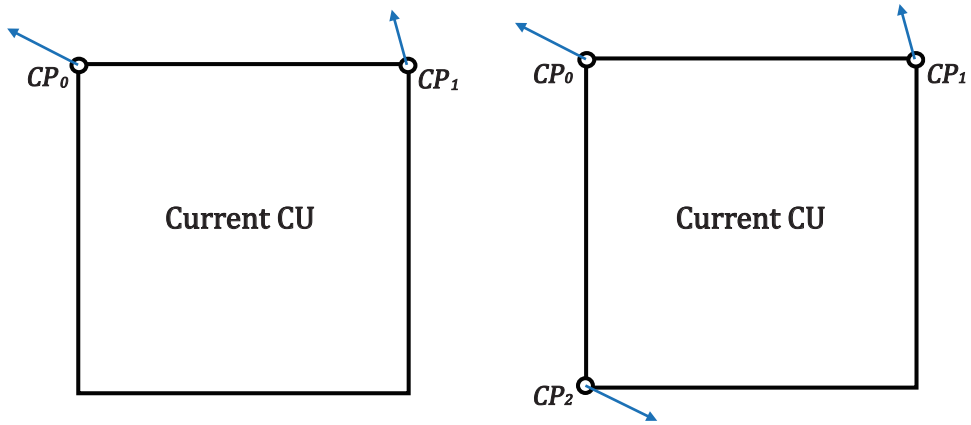


Figure 2. Affine motion vector control points: (**a**) 4-parameter motion model, (**b**) 6-parameter motion model.

To simplify the AMC, a block based AMC is applied. Figure 3 shows an example of sub block based MV derivation in a CU. The MV at the center position of each 4 × 4 sub block is derived from CPMVs and rounded to 1/16 fraction accuracy. Then the motion compensation interpolation filters are used to generate the prediction block of each sub-block with derived motion vector.

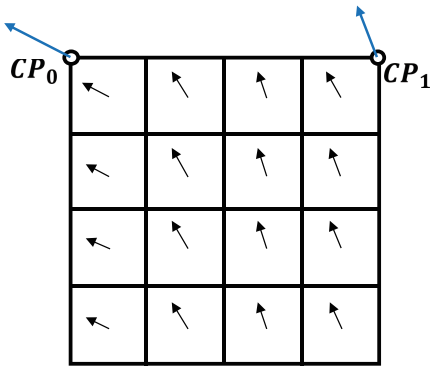


Figure 3. Affine motion vector field per sub-block.

2.2. Affine Inter Mode and Merge Mode

If a CU is coded with affine inter-mode, $\{mv_0, mv_1\}$ for 4-parameter model or $\{mv_0, mv_1, mv_2\}$ for 6-parameter model are signaled directly from the encoder to the decoder. At this moment, the difference of the CPMV of current CU and the control point motion vector prediction (CPMVP) is signaled in the bitstream. Moreover, flags for parameter type and affine mode are also signaled. Affine inter mode can be applied for CUs with both width and height larger than or equal to 16. The CPMVP candidate list size is 2 and it is derived by using the three types of CPMVP candidate generation phase in order:

1. CPMVPs extrapolated from the CPMVs of the spatial neighbour blocks

2. CPMVPs constructed using the translational MVs of the spatial neighbour blocks
3. CPMVPs generated by duplicating each of the HEVC AMVP candidates

As shown in Figure 4, neighboring blocks A, B, C, D, E, F and G are involved for generating CPMV candidate. First, if there are affine coded blocks through searching from A to G, add the CPMVs of the neighbour blocks to the CPMVP candidate list of the current CU. If the number of candidate list is smaller than 2, construct virtual CPMVP set which is composed of translational MVs $\{(mv_0, mv_1, mv_2)|mv_0 = \{mv_A, mv_B, mv_C\}, mv_1 = \{mv_D, mv_E\}, mv_2 = \{mv_F, mv_G\}, \}$. When the number of candidate list is still less than 2, finally, the list padded by the MVs composed by duplicating each of the AMVP candidates. An RD cost check process is applied to determine best CPMVP of current CU and an index indicating best CPMVP is signaled in bitstream.

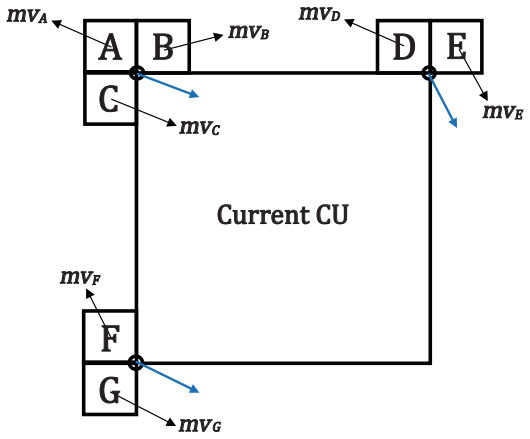


Figure 4. CPMVP candidate list for affine inter mode.

When a CU is applied in affine merge mode, the process finds the first coded block by affine mode among the neighbour candidate blocks. The selection order for the candidate block is from left, above, above right, left bottom to above as shown in Figure 5. After the CPMVs of the current CU are derived from the first neighbour block according to the affine motion model equation, the motion vector field of the current CU is generated. Like the affine inter mode, mode flag is signaled in bitstream.

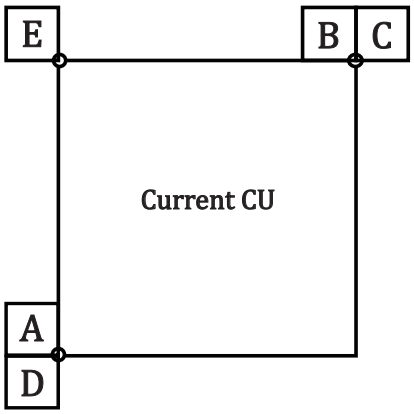


Figure 5. Candidate list for affine merge mode.

3. Proposed Perspective Affine Motion Estimation/Compensation

Affine motion estimation in the VVC is applied since it is more efficient than translational motion compensation. The coding gain can be increased by delicately estimating motion on the video sequence

in which complex motion is included. However, still it has a limit to accurately find all motions in the natural video.

Affine transformation model has properties to maintain parallelism based on the 2D plane, and thus cannot work efficiently for some sequences containing object distortions or dynamic motions such as shear and 3D affine transformation. In real world, numerous moving objects have irregular motions rather than regular translational, rotation and scaling motions. So, more elaborated motion model is needed for video coding tool to estimate motion delicately.

The basic warping transformation model can estimate motion more accurately, but this method is not suitable because of its high computational complexity and bit overhead by the large number of parameters. For these reasons, we propose a perspective affine motion compensation (PAMC) method which improve coding efficiency compared with the existing AMC method of the VVC. The perspective transformation model-based algorithm adds one more CPMV, which gives degree of freedom to all four corner vertices of the block for more precise motion vector. Furthermore, the proposed algorithm is integrated while maintaining the AMC structure. Therefore, it is possible to adopt an optimal mode between the existing encoding mode and the proposed encoding mode.

3.1. Perspective Motion Model for Motion Estimation

Figure 6 shows that the proposed perspective model with four CPs (b) can estimate motion more flexible compared with the affine model with three CPs (a). Affine motion model-based MVF of a current block is described by three CPs which are matched to $\{mv_0, mv_1, mv_2\}$ in illustration. On the other hand, one more field is added for perspective motion model-based MVF. It is composed of four CPs which are matched to $\{mv_0, mv_1, mv_2, mv_3\}$. As can be seen from Figure 6, one vertex of the block can be used additionally, so that motion estimation can be performed on various types of rectangular bases. Each side of the prediction block obtained through motion estimation based on the perspective motion model has various lengths and does not has to be parallel. The typical eight-parameter perspective motion model can be described as:

$$\begin{cases} x' = \frac{p_1 x + p_2 y + p_3}{p_7 x + p_8 y + 1}, \\ y' = \frac{p_4 x + p_5 y + p_6}{p_7 x + p_8 y + 1}. \end{cases} \tag{3}$$

where p_1, p_2, p_3, p_4, p_5, p_6, p_7 and p_8 are eight perspective model parameters. Among them, parameters p_7 and p_8 serve to give the perspective to motion model. With this characteristic, as though it is a conversion in the 2D plane, it is possible to obtain an effect that the surface on which the object is projected is changed.

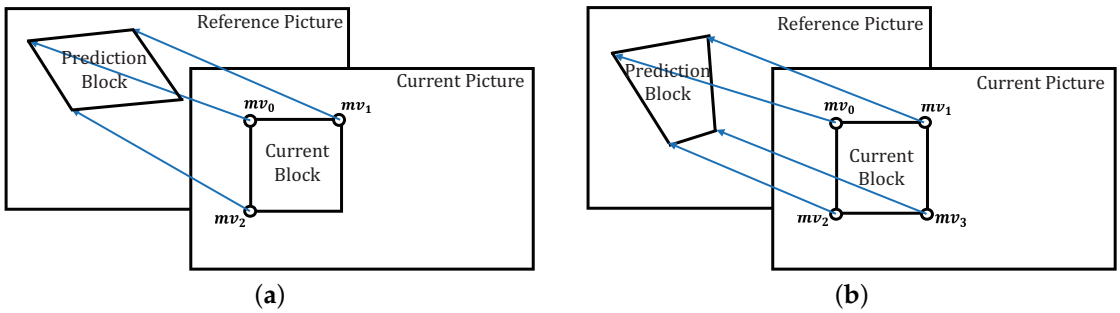


Figure 6. The motion models: (**a**) 6-parameter affine model with three CPs, (**b**) perspective model with four CPs.

Instead of these eight parameters, we used four MVs to equivalently represent the perspective transformation model like the technique applied to AMC of the existing VTM. In video codecs, using MV is more efficient in terms of coding structure and flag bits. Those four MVs can be chosen at any

location of the current block. However, in this paper, we choose the points at the top left, top right, bottom left and bottom right for convenience of model definition. In a W x H block as shown in Figure 7, we denote the MVs of $(0,0)$, $(W,0)$, $(0,H)$, and (W,H) pixel as mv_0, mv_1, mv_2 and mv_3. Moreover, we replace $p_7 \cdot W + 1$ and $p_8 \cdot H + 1$ with a_1 and a_2 to simplify the formula. The six parameters p_1, p_2, p_3, p_4, p_5 and p_6 of model can solved as following Equation (4):

$$\begin{cases} p_1 = \frac{a_1(mv_1^h - mv_0^h)}{W}, \\ p_2 = \frac{a_2(mv_2^h - mv_0^h)}{H}, \\ p_3 = mv_0^h, \\ p_4 = \frac{a_1(mv_1^v - mv_0^v)}{W}, \\ p_5 = \frac{a_2(mv_2^v - mv_0^v)}{H}, \\ p_6 = mv_0^v. \end{cases} \tag{4}$$

In addition, $p_7 \cdot W$ and $p_8 \cdot H$ can solved as Equation (5):

$$\begin{cases} p_7 \cdot W = \frac{(mv_3^h - mv_2^h)(2mv_0^v - mv_1^v) + (mv_3^v - mv_2^v)(mv_1^h - 2mv_0^h)}{(mv_3^v - mv_2^v)(mv_3^h - mv_1^h) + (mv_3^h - mv_2^h)(mv_3^v - mv_1^v)}, \\ p_8 \cdot H = \frac{(mv_3^h - mv_1^h)(2mv_0^v - mv_1^v) + (mv_3^v - mv_1^v)(mv_1^h - 2mv_0^h)}{(mv_3^v - mv_1^v)(mv_3^h - mv_2^h) + (mv_3^h - mv_1^h)(mv_3^v - mv_2^v)}. \end{cases} \tag{5}$$

Based on Equations (4) and (5), we can derive MV at sample position (x, y) in a CU by following Equation (6):

$$\begin{cases} mv^h(x,y) = \frac{\frac{a_1\left(mv_1^h - mv_0^h\right)}{W}x + \frac{a_2\left(mv_2^h - mv_0^h\right)}{H}y + mv_0^h}{\frac{a_1 - 1}{W}x + \frac{a_2 - 1}{H}y + 1}, \\ mv^v(x,y) = \frac{\frac{a_1\left(mv_1^v - mv_0^v\right)}{W}x + \frac{a_2\left(mv_2^v - mv_0^v\right)}{H}y + mv_0^v}{\frac{a_1 - 1}{W}x + \frac{a_2 - 1}{H}y + 1}. \end{cases} \tag{6}$$

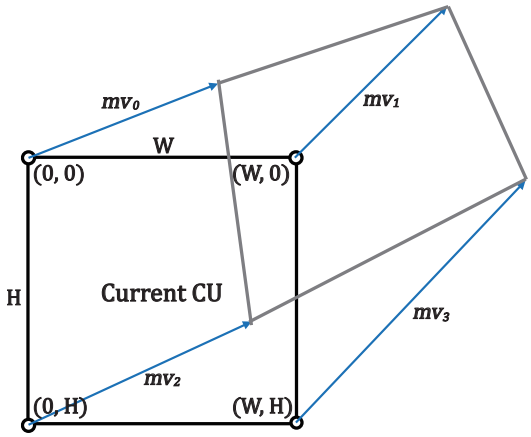


Figure 7. The representation of vertices for perspective motion model.

With the AMC, the designed perspective motion compensation also is also applied by 4×4 sub block-based MV derivation in a CU. Similarly, the motion compensation interpolation filters are used to generate the prediction block.

3.2. Perspective Affine Motion Compensation

Based on the aforementioned perspective motion model, the proposed algorithm is integrated into the existing AMC. A flowchart of the proposed algorithm is shown in Figure 8. Each motion model has its own strength. As the number of parameters increases, the precision of generating a prediction block increases. So at the same time, more bit signaling for CPMVs is required. It is effective to use the perspective motion model with four MVs is appropriate for reliability. On the other hand, if only two or three MVs are sufficient, it may be excessive to use four MVs. To take advantage of each motion model, we propose an adaptive multi-motion model-based technique.

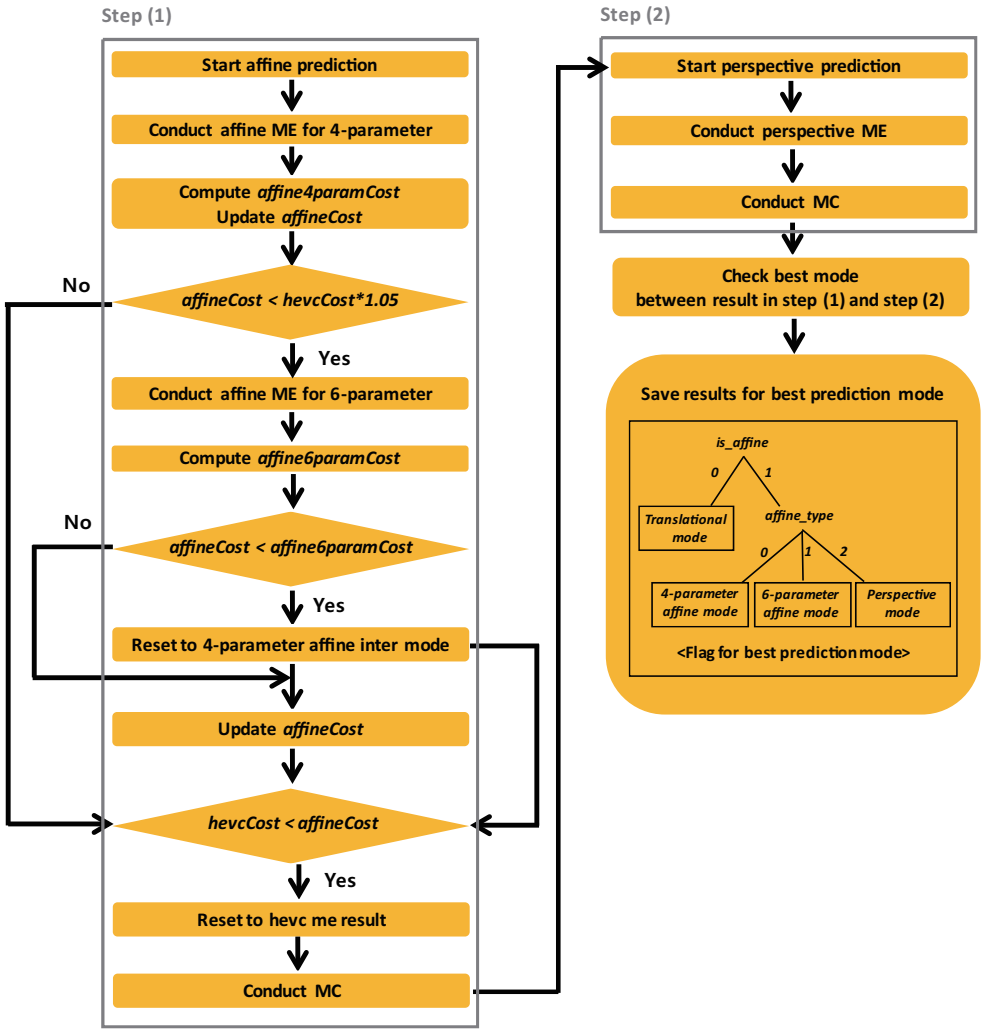


Figure 8. Flowchart of the proposed overall algorithm.

After performing fundamental translational ME and MC as in HEVC, the 4-parameter and 6-parameter affine prediction process is conducted first in step (1). Then, the proposed perspective prediction process is performed as step (2). After that, we check the best mode between result in step (1) and step (2) by RD cost check process in a current CU. Once the best mode is determined, the flag for prediction mode are signaled in the bitstream. At this time, two flags are required: affine flag and affine type flag. If the current CU is finally determined in affine mode, the affine flag is true and false otherwise. In other words, if the affine flag is false, only translational motion is used for ME. An affine type flag is signaled for a CU when its affine flag is true. When an affine type flag is 0, 4-parameter affine motion model is used for a CU. If an affine type flag is 1, 6-parameter affine motion

model-based mode is used. Finally, when an affine type flag is 2, it means that the current CU is coded in the perspective mode.

4. Experimental Results

To evaluate the performance of the proposed PAMC module, the proposed algorithm was implemented on VTM 2.0 [24]. The 14 test sequences used in the experiments were from class A to class F specified in the JVET common test conditions (CTC) [25]. Experiments are conducted under random access (RA) and low delay P (LDP) configurations and four base layer quantization parameters (QP) values of 22, 27, 32 and 37. We used 50 frames in each test sequence. The objective coding performance comparison was evaluated by the Bjontegaard-Delta Rate (BD-Rate) measurement [26]. The BD-Rate was calculated by using piece-wise cubic interpolation.

Table 1 shows the overall experimental results of the proposed algorithm for each test sequence compared with VTM 2.0 baseline. Compared with the VTM anchor, we can see that proposed PAMC algorithm can bring about 0.07% and 0.12% BD-Rate gain on average Y component in RA and LDP cases respectively, and besides it can be up to 0.45% and 0.30% on Y component for random access (RA) and low delay P (LDP) configurations, respectively. Especially in LDP, shows better gain averagely. Compared to the RA, which allow bi-directional coding schemes so have two or more prediction blocks, LDP has one predication block. For the inter prediction algorithm, the coding performance depends on the number of reference frames. For these reason, when the novel algorithm is applied to the existing encoder, the coding efficiency is higher in the LDP configuration.

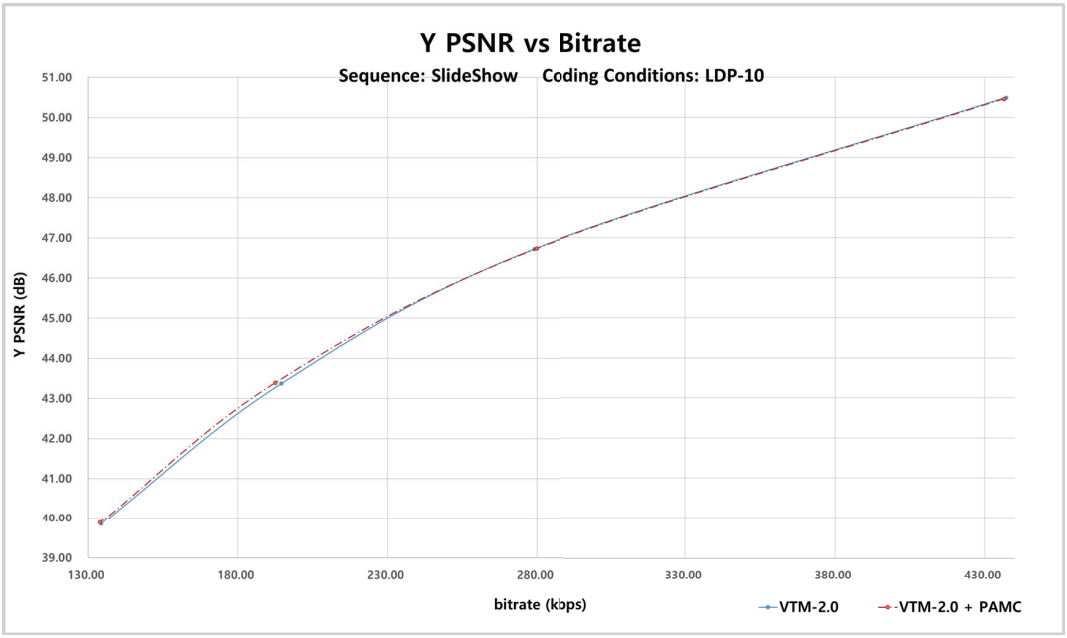
Table 1. BD-Rate (%) performance of the proposed algorithm, compared to VTM 2.0 Baseline.

Class	Sequence	Resolution	RA			LDP		
			Y	U	V	Y	U	V
A	Campfire	3840 × 2160	−0.09%	−0.02%	0.06%	-	-	-
	CatRobot1	3840 × 2160	−0.10%	0.41%	0.33%	-	-	-
B	RitualDance	1920 × 1080	−0.04%	0.15%	0.22%	−0.06%	0.55%	0.01%
	BasketballDrive	1920 × 1080	−0.14%	−0.05%	0.44%	−0.06%	0.28%	0.47%
	BQTerrace	1920 × 1080	−0.08%	−0.11%	−0.03%	−0.08%	0.38%	−0.16%
C	BasketballDrill	832 × 480	−0.09%	−0.12%	−0.16%	−0.02%	0.04%	−0.15%
	PartyScene	832 × 480	−0.06%	−0.21%	−0.04%	0.01%	0.46%	0.24%
D	BQSquare	416 × 240	0.03%	0.45%	−0.04%	−0.03%	−0.42%	−1.87%
	RaceHorses	416 × 240	0.07%	−0.76%	0.09%	−0.25%	0.21%	0.49%
E	FourPeople	1280 × 720	-	-	-	−0.16%	−0.58%	0.15%
	KristenAndSara	1280 × 720	-	-	-	0.07%	−0.22%	0.03%
F	BasketballDrillText	832 × 480	−0.01%	−0.18%	0.25%	−0.07%	−0.57%	−0.28%
	SlideEditing	1280 × 720	−0.07%	−0.03%	−0.03%	−0.28%	−0.22%	−0.47%
	SlideShow	1280 × 720	−0.30%	1.29%	0.19%	−0.45%	−1.39%	0.08%
Avg.			**−0.07%**	**0.07%**	**0.11%**	**−0.12%**	**−0.12%**	**−0.12%**

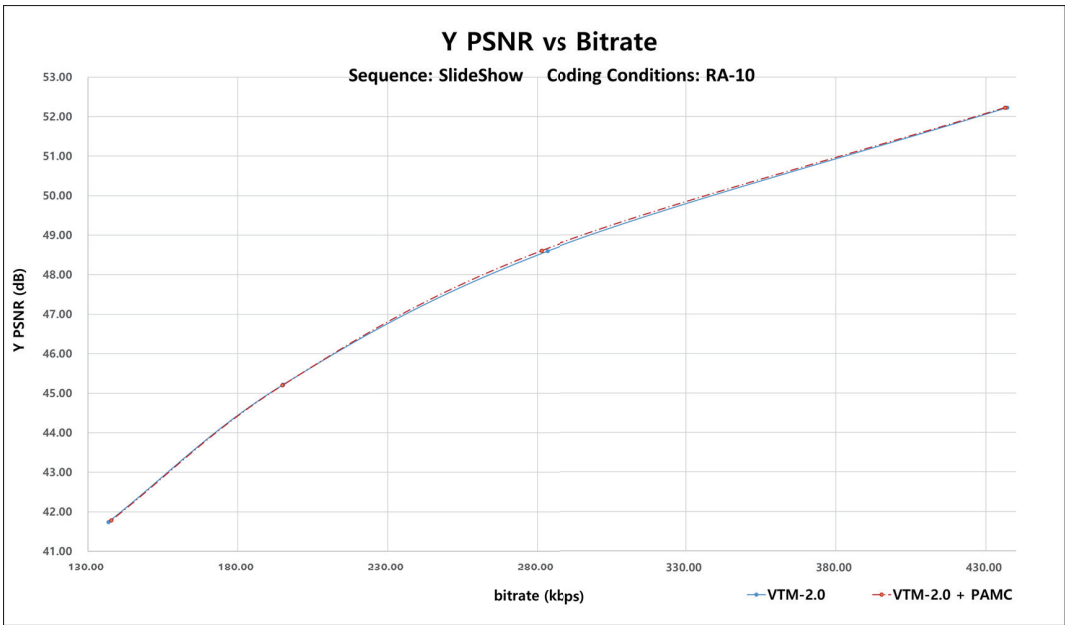
Although there were a small BD-rate losses of chroma components, the chroma components are usually less sensitive to the perception of human eye for recognition. So the luminance component is more important to measure the performance. The proposed algorithm achieved up to 0.45% and in the most of sequences, the coding gain was obtained in Y component (luminance component) although they were small value in some sequences such as BasketballDrill, PartyScene, and BQsquare with RA configuration. For RaceHorse, SlideEditing, and FourPeople sequences in LDP configuration, 0.25%, 0.28%, and 0.45% of DB-rate savings of Y component were observed in Table 1.

Some examples of rate-distortion (R-D) curves are shown in Figure 9. The R-D curves also verify that the proposed perspective affine MC can achieve better coding performance compared with the VTM baseline. It can be seen from Figure 9 that the proposed algorithm works more efficiently than

the existing affine MC algorithm in both the LDP and RA configurations. For LDP coding condition, it is more efficient in $QP = 32$ and $QP = 37$ cases and for RA coding condition, it seems to have an effect on $QP = 22$ and $QP = 27$.



(**a**) SlideShow in LDP mdoe.



(**b**) SlideShow in RA mdoe.

Figure 9. The R-D curves of the proposed perspective affine MC framework.

To improve the coding efficiency through the proposed algorithm, the test sequence have to contain irregular object distortions or dynamic rapid motions. Figure 10 shows the examples of perspective motion area in test sequences. Figure 10a–c present the examples of sequence "Campfire", "CatRobot1" and "BQTerrace", respectively. The class A sequence "Campfire" contains a bonfire that moves inconsistently and steadily. Also "CatRobot1" contains a lot of flapping scarves and a flipped book, and class B sequence "BQTerrace" involves the ripples on the surface of the water. All of such moving objects commonly include object distortions whose shape changes. Because of

this, those sequences can be compressed more efficiently by the proposed framework. The results of "Campfire" and "CatRobot1" sequences show that proposed PAMC achieves 0.09% and 0.10% BD-Rate savings respectively on Y component in RA. The result of "BQTerrace" sequence shows a coding gain of 0.08% on Y component for both RA and LDP.

Figure 11 shows an example of comparing AMC in VVC and the proposed PAMC using "CatRobot1" sequence. It is a result for POC 24 encoded by setting QP 22 in RA. Figure 11a presents the coded CU with affine mode in VVC baseline and Figure 11b shows the coded CU with affine and perspective mode in the proposed framework. If the unfilled rectangles imply the CUs coded in affine mode and the filled rectangles imply the CUs coded in perspective mode, in Figure 11b, some filled rectangles can be found on scarves and on the pages of a book. The class B sequences "RitualDance" and "BasketballDrive" and class C sequence "BasketballDrill", which contain dynamic fast movements, can also be seen to bring coding gains in both RA and LDP configurations. For the three sequences mentioned above, performance result shows that proposed PAMC results in 0.04%, 0.14%, and 0.09% BD-Rate savings respectively on Y component in RA. In LDP configuration, BD-Rate gains are 0.06%, 0.06% and 0.02% respectively on Y component.

In class F which has the screen content sequences, rapid long range motions as large as half a frame often happen like browsing and document editing. Even in this case, the proposed PAMC algorithm can bring BD-Rate gain. In particular, the "SlideShow" sequence gives the largest coding gain resulting in 0.30% and 0.45% BD-Rate savings on Y component in RA and LDP respectively. Besides, on U component in LDP, it brings 1.39% of BD-Rate gain.

When the resolution of sequence is too small such as class D, the sequence contains a small amount of textures and object content. Therefore, it is possible to estimate the motion accurately with only using further enhanced configuration. For that reason, it can be seen from the result of class D that proposed algorithm contributes to overall coding gain in LDP but not in RA. The result of "BQSquare" sequence shows that proposed PAMC achieves 0.03%, 0.42% and 1.87% BD-Rate savings on Y, U and V components respectively in LDP. For "RaceHorces" sequence, the result shows 0.25% of BD-Rate gain on Y component in LDP.

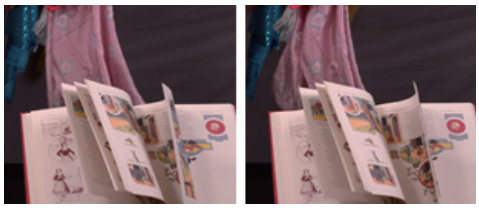
As the proposed algorithm is designed to better describe the motion with distortion of object shape, some equirectangular projection (ERP) format sequences [27] are selected to verify the performance of the proposed algorithm. Figure 12 shows an example of ERP sequence. Figure 12a presents a frame of the "Broadway" sequence and Figure 12b shows a enlarged specific area of the frame. Figure 12c shows a picture in posterior frame for the same area. The ERP produces significant deformation, especially in the pole area. It can be obviously seen from Figure 12 that the distortion of the building object occurs. Perspective motion model can take such deformation into account when compressing the planar video of panoramic content.

The R-D performance of the proposed algorithm for the ERP test sequences is illustrated in Table 2. From Table 2, we can see that the proposed framework can be up to 0.15% on Y component for low delay P (LDP) configuration. The experimental results obviously demonstrate that the proposed perspective affine motion model can well represent the motion with distortion of object shape.

As shown in some video coding research [28,29], the results show 0.07% and 0.12% BD-Rate gain on average, respectively. Furthermore, several advanced affine motion estimation algorithms in JVET meeting documents [30–32], the results show 0.09%, 0.09% and 0.13% BD-Rate gain on average Y component. Compared with these results, the performance of the proposed algorithm is also competitive. Moreover, our proposed method contributes in that the encoder can be more robust in natural videos by proposing a flexible motion model for affine ME, one of the main inter prediction tools of the existing VTM codec.



(**a**) Campfire.



(**b**) CatRobot1.

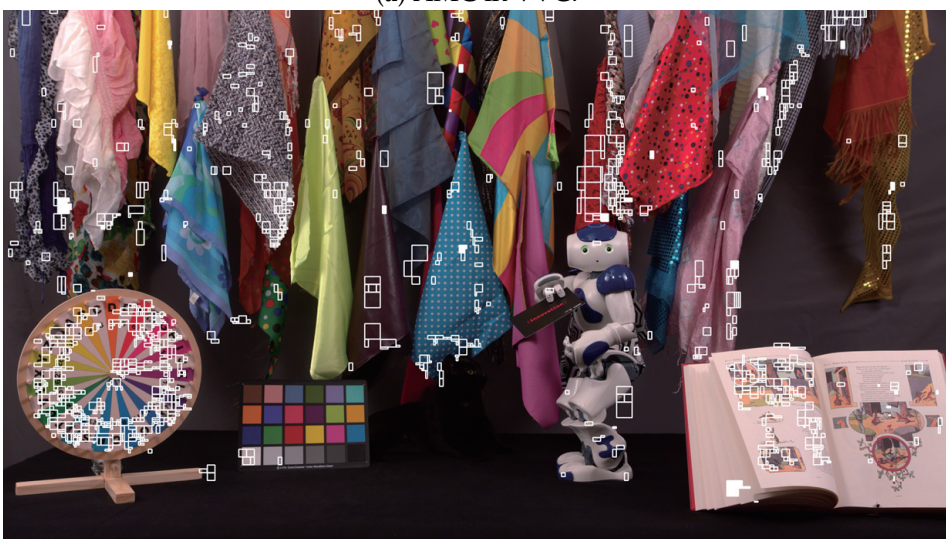


(**c**) BQTerrace.

Figure 10. An examples of perspective motion area in test sequences.



(**a**) AMC in VVC.



(**b**) Proposed PAMC.

Figure 11. An example of CUs with affine or perspective motion, CatRobot1, RA, QP22, POC24.

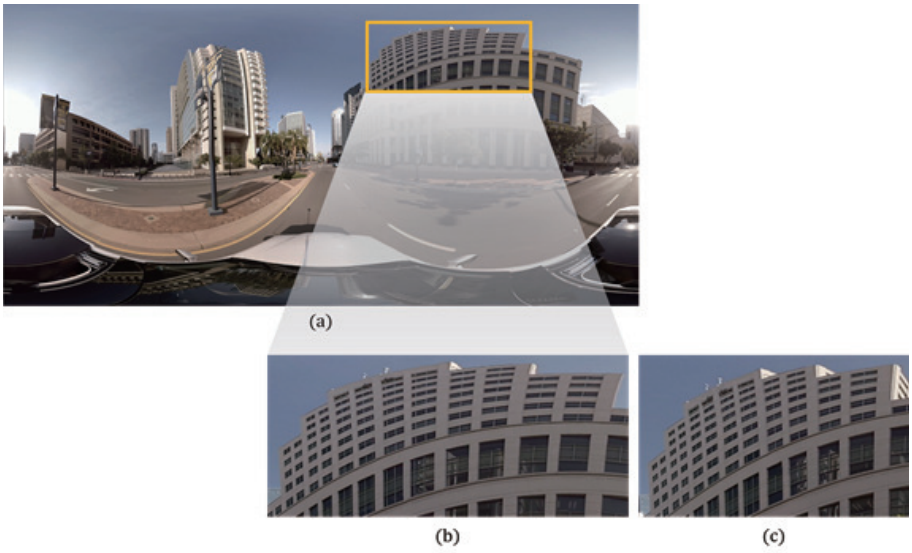


Figure 12. An example of the ERP format sequence (Broadway).

From experimental results, the designed PAMC achieved better coding gain compared to VTM 2.0 Baseline. It means that the proposed PAMC scheme can be applied for providing better video quality in terms of a limited network bandwidth environment.

Table 2. BD-Rate (%) performance of the proposed algorithm for ERP format sequences compared to VTM 2.0 Baseline.

Sequence	Resolution	LDP		
		Y	U	V
Broadway	6144 × 3072	−0.15%	0.03%	−0.19%
Freefall	6144 × 3072	−0.13%	0.12%	0.35%
BranCastle2	6144 × 3072	−0.08%	−0.11%	−0.28%
Balboa	6144 × 3072	−0.10%	−0.06%	0.17%
Avg.		**−0.12%**	**−0.01%**	**0.01%**

5. Conclusions

In this paper, an efficient perspective affine motion compensation framework was proposed to estimate further complex motions beyond the affine motion. Affine motion model has properties which maintains parallelism, and thus cannot work efficiently for some sequences containing object distortions or rapid dynamic motions. In the proposed algorithm, an eight-parameter perspective motion model was first defined and analyzed. Like the technique applied to AMC of existing VTM, we designed four MVs based motion model instead of using eight parameters. Then the perspective motion model-based motion compensation algorithm was proposed. To take advantage of each affine and perspective motion model, we proposed an adaptive multi-motion model-based technique. The proposed framework was implemented in the reference software of VVC. We experimented with two kinds of sequences. In addition to experimenting with JVET common test condition sequences, we demonstrated the effectiveness of the proposed algorithm by showing the results for the equirectangular projection format sequences. The experimental results showed that the proposed perspective affine motion compensation framework could achieve much better BD-Rate performance compared with the VVC baseline especially for sequences that contain irregular object distortions or dynamic rapid motions.

Although the proposed algorithm improved the inter-prediction of the VVC video standard technology, there is still room for further improvement. For future studies, the higher-order motion

models should be investigated and applied for three-dimensional modeling of motion. Higher-order models can improve the accuracy of irregular motion, but they can result in an increase in the number of bits, as these parameters must be sent together. Considering these points, an approximation model should also be conducted to be compatible for the VVC standard.

Author Contributions: Conceptualization, B.-G.K.; methodology, Y.-J.C.; software, Y.-J.C.; validation, D.-S.J.; formal analysis, W.-S.C.; Writing—Original Draft preparation, Y.-J.C.; Writing—Review and Editing, B.-G.K.; supervision, B.-G.K.;funding acquisition, D.-S.J.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B04934750) and partially supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by ICT R&D program of MSIT/IITP [No. 2017-0-00072, Development of Audio/Video Coding and Light Field Media Fundamental Technologies for Ultra Realistic Tera-media.

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

References

1. Draft ITU-T Recommendation and final draft international standard of joint video specification (ITU-T Rec. H. 264 | ISO/IEC 14496-10 AVC). In Proceedings of the 5th Meeting, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Geneva, Switzerland, 9–17 October 2002.
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, B.; Chen, J.; Liu, S. *Versatile Video Coding (Draft 2)*; JVET-K1001; JVET: Ljubljana, Slovenia, 2018.
4. Chen, J.; Alshina, E. *Algorithm Description for Versatile Video Coding and Test Model 1 (VTM 1)*; JVET-J1002; JVET: San Diego, CA, USA, 2018.
5. Chen, J.; Ye, Y.; Kim, S. *Algorithm Description for Versatile Video Coding and Test Model 2 (VTM 2)*; JVET-K1002; JVET: Ljubljana, Slovenia, 2018.
6. Seferidis, V.; Ghanbari, M. General approach to block-matching motion estimation. *Opt. Eng.* **1993**, *32*, 1464–1474. [CrossRef]
7. Lee, O.; Wang, Y. Motion compensated prediction using nodal based deformable block matching. *J. Vis. Commun. Image Represent.* **1995**, *6*, 26–34. [CrossRef]
8. Cheung, H.K.; Siu, W.C. Local affine motion prediction for H.264 without extra overhead. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Paris, France, 30 May–2 June 2010; pp. 1555–1558.
9. Kim, H.-S.; Lee, J.-H.; Kim, C.-K.; Kim, B.-G. Zoom motion estimation using block-based fast local area scaling. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1280–1291. [CrossRef]
10. Narroschke, M.; Swoboda, R. Extending HEVC by an affine motion model. In Proceedings of the 2013 Picture Coding Symposium (PCS), San Jose, CA, USA, 8–11 December 2013; pp. 321–324.
11. Huang, H.; Woods, J.W.; Zhao, Y.; Bai, H. Affine SKIP and DIRECT modes for efficient video coding. In Proceedings of the Visual Communications and Image Processing (VCIP), San Diego, CA, USA, 27–30 November 2012; pp. 1–6.
12. Heithausen, C.; Vorwerk, J.H. Motion compensation with higher order motion models for HEVC. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19–24 April 2015; pp. 1438–1442.
13. Chen, H.; Liang, F.; Lin, S. Affine SKIP and MERGE modes for video coding. In Proceedings of the 2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), Xiamen, China, 19–21 October 2015; pp. 1–5.
14. Heithausen, C.; Bläser, M.; Wien, M.; Ohm, J.R. Improved higher order motion compensation in HEVC with block-to-block translational shift compensation. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2008–2012.
15. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]

16. Heithausen, C.; Meyer, M.; Bläser, M.; Ohm, J.R. Temporal prediction of motion parameters with interchangeable motion model. In Proceedings of the 2017 Data Compression Conference (DCC), Snowbird, UT, USA, 4–7 April 2017; pp. 400–409.
17. Li, L.; Li, H.; Lv, Z.; Yang, H. An affine motion compensation framework for High Efficiency Video Coding. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 24–27 May 2015; pp. 525–528.
18. Li, L.; Li, H.; Liu, D.; Li, Z.; Yang, H.; Lin, S.; Chen, H.; Wu, F. An Efficient Four-Parameter Affine Motion Model for Video Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 1934–1948. [CrossRef]
19. Lin, S.; Chen, H.; Zhang, H.; Maxim, S.; Yang, H.; Zhou, J. *Affine Transform Prediction for Next Generation Video Coding*; ITU-T SG16 Doc. COM16–C1016; Huawei Technologies: Shenzhen, China, 2015.
20. Chen, J.; Alshina, E.; Sullivan, G.J.; Ohm, J.R.; Boyce, J. *Algorithm Description of Joint Exploration Test Model 1;* JVET-A1001; JVET: Geneva, Switzerland, 2015.
21. Choi, Y.J.; Kim, J.H.; Lee, J.H.; Kim, B.G. Performance Analysis of Future Video Coding (FVC) Standard Technology. *J. Multimed. Inf. Syst.* **2017**, *4*, 73–78.
22. Zhang, K.; Chen, Y.W.; Zhang, L.; Chien, W.J.; Karczewicz, M. An Improved Framework of Affine Motion Compensation in Video Coding. *IEEE Trans. Image Process.* **2019**, *28*, 1456–1469. [CrossRef] [PubMed]
23. Lichtenauer, J.F.; Sirmacek, B. A semi-automatic procedure for texturing of laser scanning point clouds with google streetview images. Available online: https://repository.tudelft.nl/islandora/object/uuid%3A8bb4d40b-0950-471f-b774-7f74449fe26e (accessed on 16 September 2019).
24. Versatile Video Coding (VVC) Test Model 2.0 (VTM 2.0). Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.git (accessed on 16 September 2018).
25. Bossen, F.; Boyce, J.; Suehring, K.; Li, X.; Seregin, V. *JVET Common Test Conditions and Software Reference Configurations for SDR Video*; JVET-K1010; JVET: Ljubljana, Slovenia, 2018.
26. Bjøntegaard, G. *Calculation of Average PSNR Differences between RDcurves*; ITU-T SG.16 Q.6, Document VCEG-M33; ITU-T VCEG: Austin, TX, USA, 2001.
27. Hanhart, P.; Boyce, J.; Choi, K. *JVET Common Test Conditions and Evaluation Procedures for 360 Video*; JVET-K1012: Ljubljana, Slovenia, 2018.
28. Yoon, Y.U.; Kim, H.H.; Lee, Y.J.; Kim, J.G. Methods of padding inactive regions for rotated sphere projection of 360 video. In Proceedings of the 2019 Joint International Workshop on Advanced Image Technology (IWAIT) and International Forum on Medical Imaging in Asia (IFMIA), Singapore, 6–9 January 2019.
29. Ma, S.; Lin, Y.; Zhu, C.; Zheng, J.; Yu, L.; Wang, X. Improved segment-wise DC coding for HEVC intra prediction of depth maps. In Proceedings of the Signal and Information Processing Association Annual Summit and Conference (APSIPA), Siem Reap, Cambodia, 9–12 December 2014.
30. Zhao, J.; Kim, S. H.; Li, G.; Xu, X.; Li, X.; Liu, S. *CE2: History Based Affine Motion Candidate (Test 2.2.3)*; JVET-M0125; JVET: Marrakech, Morocco, 2019.
31. Galpin, F.; Robert, A.; Le Léannec, F.; Poirier, T. *CE2.2.7: Affine Temporal Constructed Candidates*; JVET-M0256; JVET: Marrakech, Morocco, 2019.
32. Zhang, K.; Zhang, L.; Liu, H.; Xu, J.; Wang, Y.; Zhao, P.; Hong, D. *CE2-Related: History-Based Affine Merge Candidates*; JVET-M0266; JVET: Marrakech, Morocco, 2019.



© 2019 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 (http://creativecommons.org/licenses/by/4.0/).