

## 다목적 비디오 부호화를 위한 고속 어파인 움직임 예측 방법

### Fast Affine Motion Estimation Method for Versatile Video Coding

정승원<sup>1</sup>, 전동산<sup>2\*</sup>

Seong-Won Jung<sup>1</sup>, Dong-San Jun<sup>2\*</sup>

#### 〈Abstract〉

Versatile Video Coding (VVC) is the most recent video coding standard, which had been developed by Joint Video Expert Team (JVET). It can improve significant coding performance compared to the previous standard, namely High Efficiency Video Coding (HEVC). Although VVC can achieve the powerful coding performance, it requires the tremendous computational complexity of VVC encoder. Especially, affine motion compensation (AMC) was adopted the block-based 4-parameter or 6-parameter affine prediction to overcome the limit of translational motion model while VVC require the cost of higher encoding complexity. In this paper, we proposed the early termination of AMC that determines whether the affine motion estimation for AMC is performed or not. Experimental results showed that the proposed method reduced the encoding complexity of affine motion estimation (AME) up to 16% compared to the VVC Test Model 17 (VTM17).

*Keywords : Versatile Video Coding, High Efficiency Video Coding, Affine Motion Estimation, Fast Encoding, Rate Distortion*

---

1 정회원, 동아대학교 컴퓨터공학과,  
부산광역시 사하구 낙동대로 550번길 37  
E-mail: swjung@donga.ac.kr

2\* 교신저자, 동아대학교 컴퓨터공학과, 교수,  
부산광역시 사하구 낙동대로 550번길 37  
E-mail: dsjun@dau.ac.kr

1 Dept. of Computer Engineering, Dong-A University,  
Nakdong-daero 550 beon-gil, Saha-gu, Busan, Korea  
E-mail: swjung@donga.ac.kr

2\* Corresponding Author, Dept. of Computer Engineering,  
Dong-A University, Professor, Nakdong-daero 550 beon-gil,  
Saha-gu, Busan, Korea  
E-mail: dsjun@dau.ac.kr

## 1. 서론

최근 UHD급 (Ultra HD) 고화질의 초실감 미디어 서비스가 빠르게 확산될 뿐만 아니라 VR (Virtual Reality)/AR (Argumented Reality)과 같은 몰입형 비디오 서비스가 제공되고 있다. 새로운 시청경험을 제공해 줄 수 있는 초실감미디어 콘텐츠들은 대용량의 원본 데이터량을 가지기 때문에 이러한 대용량 데이터를 네트워크 대역폭에 맞춰 효율적으로 전송하기 위해서는 높은 압축 성능을 갖는 비디오 코덱의 필요성이 지속적으로 제기되고 있다. ISO/IEC MPEG (Moving Picture Experts Group)과 ITU-T VCEG (Video Coding Experts Group)은 2018년 JVET을 결성하여 종래의 압축 표준인 HEVC[1]보다 2배의 압축 효율을 갖는 VVC[2] 표준화를 2020년 상반기에 완료하였다. VVC는 HEVC 보다 높은 압축성능을 달성하기 위해 QTMTT (Quaternary Tree plus Multi-Type Tree) 기반의 부호화 방법을 채택하여 기존 대비 유연해진 블록 분할 구조를 갖게 되었을 뿐만 아니라, 예측효율을 높이기 위해 다수의 화면내 및 화면간 예측기술을 신규로 채택하였다. 특히 화면간 예측에서 채택된 기법으로는 AMVR (Adaptive Motion Vector Resolution), Affine 예측모드 등이 추가되었고, Merge 모드로는 CIIP (Combined Intra and Inter Prediction), GPM (Geometric Partition Mode), MMVD (Merge with MVD), DMVR (Decoder Motion Vector Refinement) 과 같은 툴들이 추가되었다. 신규로 채택된 화면간 예측기술 중에서, Affine 예측모드는 평행이동, 회전, 확대, 축소, 전단이동과 같은 움직임을 효과적으로 예측하기 위해 채택된 기법이다. 종래의 압축 표준인 HEVC에서는 AMVP (Advanced Motion Vector Prediction)를 적용하여 오직 평행이동하는 물체만 예측할 수 있었던 반면, 회전하거나 전단이

동과 같은 복잡한 움직임을 예측하는데 한계를 가지고 있었다. Affine 예측 방법은 이처럼 영상의 특성에 따라 회전, 확대, 축소 등에 대한 예측을 효과적으로 수행할 수 있는 기술이지만, 기존 화면간 예측모드에 비해 높은 부호화 복잡도를 필요로 하게 되었다. 예를 들어, Affine 예측 모드의 Tool-off 테스트를 통하여 전체 부호화기의 계산 복잡도를 20% 차지함을 알 수 있다[3]. 따라서 모바일 환경과 같이 제한된 메모리 사양을 갖거나 연산성능이 낮은 플랫폼에서는 대용량의 UHD급 고해상도 서비스를 실시간 혹은 고속 부호화 하기 위해 비디오 코덱의 복잡도를 감소시킬수 있는 고속화 알고리즘 적용이 필수적으로 요구된다.

본 논문의 구성은 다음과 같다. VVC의 Affine 모드에 대하여 2장에서 소개한 후, 3장에서는 제안하는 고속 Affine 결정 알고리즘에 대하여 설명한다. 4장에서 제안 방법에 대한 결과를 보여준 후, 5장에서 결론을 맺는다.

## 2. 기존 방법

VVC 표준에서 제공하는 Affine 예측방법은 4-parameter 및 6-parameter affine 모델을 이용하여 4x4 서브 블록 기반의 화면간 예측을 수행한다[3]. Fig. 1(a)와 같이 4-parameter affine 모델에서는 평행이동이 아닌 회전, 확대, 축소와 같은 화면간 움직임을 예측할 수 있으며, Fig. 1(b)의 6-parameter affine 모델은 회전, 확대, 축소 뿐만 아니라 전단이동과 같은 화면간 움직임을 예측할 수 있도록 설계되었다. 이때, 4-parameter affine 모델의 움직임 벡터  $MV (mv^h, mv^v)$ 는 식 (1)과 같이 유도되며,

$$\begin{cases} mv^h(x,y) = a(x-x_0) - b(y-y_0) + mv_0^h \\ mv^v(x,y) = b(x-x_0) + a(y-y_0) + mv_0^v \end{cases} \quad (1)$$

6-parameter affine 모델의 움직임 벡터 MV ( $mv^h, mv^v$ )는 식 (2)와 같이 유도된다.

$$\begin{cases} mv^h(x,y) = a(x-x_0) - c(y-y_0) + mv_0^h \\ mv^v(x,y) = b(x-x_0) + d(y-y_0) + mv_0^v \end{cases} \quad (2)$$

상기 식 (1), (2)에서 가중치에 해당하는 ( $a, b, c, d$ ) 매개변수는 식 (3)과 같이 계산된다.

$$\begin{aligned} a &= \frac{(mv_1^h - mv_0^h)}{W}, b = \frac{(mv_1^v - mv_0^v)}{W}, \\ c &= \frac{(mv_2^h - mv_0^h)}{H}, d = \frac{(mv_2^v - mv_0^v)}{H}, \end{aligned} \quad (3)$$

Fig. 1에서  $mv_0, mv_1$  및  $mv_2$ 는 Control Point Motion Vector (CPMV)로서 좌측상단 ( $x_0, y_0$ ),

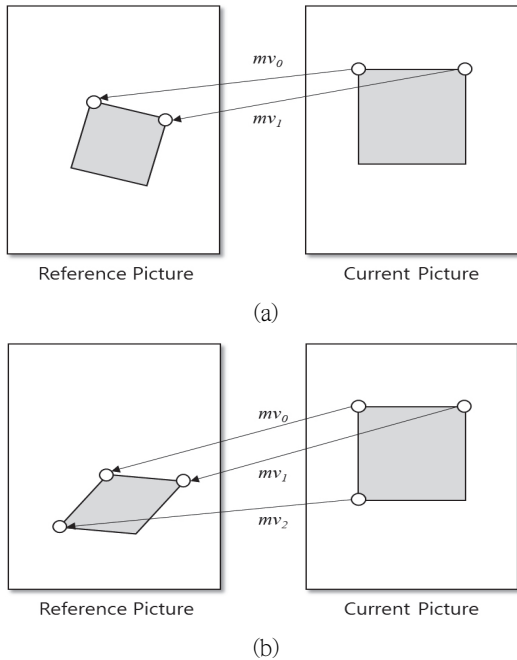


Fig. 1 The affine motion models: (a) 4-parameter affine model with two CPMVs, (b) 6-parameter affine model with three CPMVs

우측상단 ( $x_0 + W, y_0$ ), 좌측하단 ( $x_0 + W, y_0 + H$ )의 움직임 정보를 각각 의미하며, 이때  $W$  와  $H$ 는 현재 부호화되는 Coding Unit (CU)의 가로 및 세로 크기를 나타낸다.

VVC는 AMC과정에서 4x4 서브블록 단위로 Motion estimation 및 Motion compensation (ME/MC) 과정을 수행함에 따라, Fig. 2에 도시된 바와 같이, 식 (1) 또는 (2)를 통해 현재 CU를 4x4 서브블록으로 분할하여 서브블록당 MV를 유도한다[4]. 참고로 VVC 표준에서는 Affine 예측방법은 휘도 및 색차 블록 모두에 적용될 수 있다.

VVC에서는 Affine inter 모드와 Affine merge 모드가 채택되어 있으며, 각각 명시적인 방식과 묵시적인 방식으로 인코더에서 디코더로 CPMV를 전송한다. Affine inter 모드로 부호화된 경우, 4-parameter 또는 6-parameter affine 모드 식별에 필요한 플래그가 전송된 이후, 2개 또는 3개의 CPMV가 전송된다[5]. 추가적으로, VVC는 시그널링된 CPMV의 해상도가 CU 레벨에서 1/16, 1/4 혹은 1 픽셀 단위로 전송될 수 있도록 설계되었다.

Affine 모드의 후보 리스트를 생성하기 위해서, 이웃한 Affine 모드에서 CPMV를 상속하거나 이웃 블록의 MV에서 CPMV를 도출할 수 있다. 상

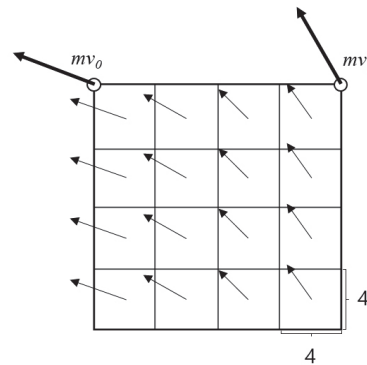


Fig. 2 Affine motion vector field in a 16x16 block coded with AMC

속된 Affine 후보는 Affine 모드로 코딩된 이웃 블록에서만 CPMV를 도출할 수 있다. Fig. 3에 도시된 바와 같이 이웃 블록 A0는 6-parameter affine 모델로 코딩되어 CPMV의  $\{mv_0^{A0}, mv_1^{A0}, mv_2^{A0}\}$ 를 갖는 CU이다. 만약  $\{mv_0^C, mv_1^C, mv_2^C\}$ 가 Current CU의 CPMV라고 가정하면  $mv_0^C$ 는  $mv_0 = mv_0^{A0}, mv_1 = mv_1^{A0}, mv_2 = mv_2^{A0}$  및  $(x, y) = (x_0 - x'_0, y_0 - y'_0)$  이며,  $mv_1^C$  및  $mv_2^C$ 는 동일한 방식으로 유도되지만  $(x, y)$ 는

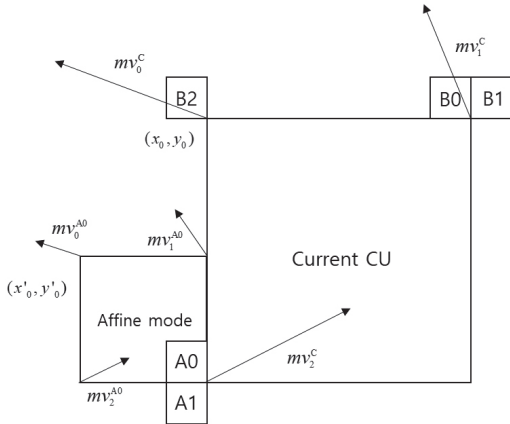


Fig. 3 Inherited affine prediction candidates

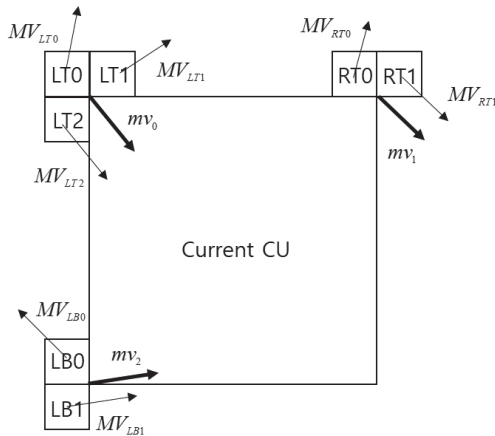


Fig. 4 Constructed affine prediction candidates

각각  $(x_1 - x'_0, y_1 - y'_0)$  및  $(x_2 - x'_0, y_2 - y'_0)$ 이다. Constructed affine 후보는 이웃 블록이 Affine으로 코딩되지 않은 경우에도 생성할 수 있다. Fig. 4는 Constructed affine 후보를 구성하는 방법의 예를 보여준다. 3개의 가상 CPMV  $\{mv_0, mv_1, mv_2\}$ 은 화면간 예측으로 코딩된 주변의 이웃 블록에서 가장 먼저 유도되는 MV로 설정된다. 예를 들어, 이웃한 블록 LT0, LT1, LT2를 차례로 체크하여 화면간 예측기법으로 코딩된 첫 번째 블록의 MV를  $mv_0$ 로 설정한다. 이후  $mv_1$ 은 RT0, RT1을 순서대로 체크하여 첫 번째로 확인되는 MV를  $mv_1$ 으로 설정하며,  $mv_2$ 도 동일한 방법으로 LB0, LB1 순으로 체크하여 MV를 설정한다. 만약 MV가 존재하지 않는다면, 이웃한 하나의 MV만 사용하여 동일한 MV로 CPMV 후보 리스트를 생성하게 되며, 근처에 MV가 존재하지 않은 경우 Zero MV로 CPMV 후보 리스트를 구성한다. 상술한 방식으로 CPMV 후보 리스트를 최대 2개 까지 생성하여 AME를 수행하여 최적의 Affine inter 모드를 도출한다.

### 3. 제안 방법

VVC에서는 Current CU의 최적 부호화 모드를 찾기 위해서 모든 예측 모드에 대한 율-왜곡 최적화 (Rate Distortion Optimization, RDO) 과정을 수행한다. 특히 Affine inter 모드는 다른 예측 모드들에 비해 많은 부호화 복잡도를 차지하므로 본 논문에서는 RDO 과정에서 Affine inter 모드를 조기종료할 수 있는 고속화 알고리즘을 제안한다.

Fig. 5에서 도시된 바와 같이 제안하는 고속 결정 알고리즘을 적용하기 위해 Parent CU 및 Current CU에 대하여 정의하였다. 본 논문에서 정

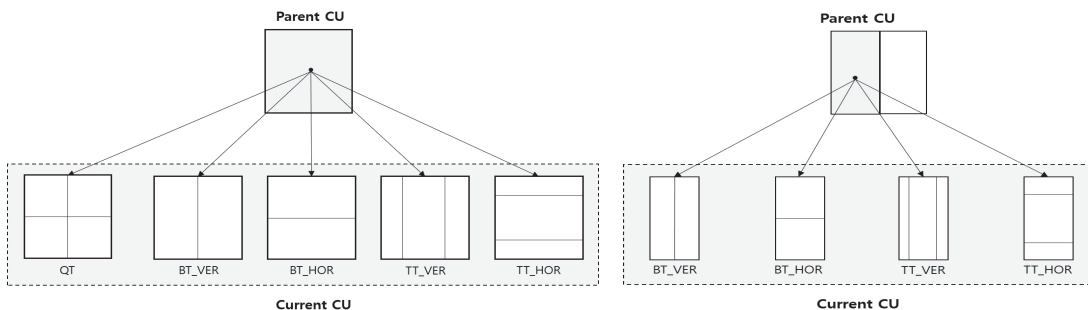


Fig. 5 Definition of parent CU and Current CU. (a) Example of QT as a parent CU and (b) Example of MTT as a parent CU

```

Full-RD perform for affine merge mode
Full-RD perform for merge mode
Full-RD perform for GPM mode
if Caffine_merge && !CAMVR_integer then
    affine motion estimation E/T of integer-pel
else affine motion estimation
endif
if Paffine4p then
    6-parameter affine motion estimation E/T
else 6-parameter affine motion estimation
endif
    
```

Fig. 6 Fast Affine Algorithm

의한 Parent CU는 정사각형 형태의 QT 노드이거나 혹은 Current CU 영역을 포함하는 직사각형 혹은 정사각형 형태의 MTT 노드일 수 있다. 예를 들어, Fig. 5(a)와 같이 분할된 Current QT, BT 및 TT CU는 2Nx2N의 동일한 Parent CU를 가지며, 이와 유사하게 Fig. 5(b)와 같이 분할된 BT 및 TT CU는 해당 CU를 포함하는 2NxN의 Parent CU를 가질 수 있도록 정의하였다.

Fig. 6은 제안하는 조기종료 알고리즘에 대한 Pseudo Code를 보여준다. 여기서 C, P 및 “!”는 Current CU, Parent CU 및 NOT의 논리 연산자로 각각 표시된다. 그리고 Current CU의 인코딩 과정에서 최적의 모드를 아래 첨자로 표시하였으며, affine4p는 6-parameter affine 모드와 4-parameter affine 모드를 비교하여 최적의 모드가 4-parameter

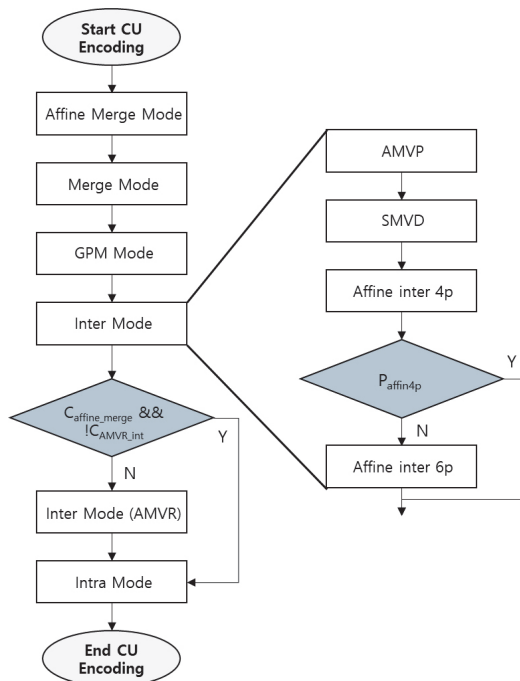


Fig. 7 Proposed fast AME encoding process

affine 모드로 결정된 것을 의미한다. 제안된 알고리즘은 Parent CU와 Current CU에서 유도되는 정보를 통하여 CPMV를 유도하는 AME 과정을 조기종료할 수 있도록 설계하였다. Fig. 7은 VTM 내에 제안 방법이 적용된 CU 부호화 과정을 나타내며, 본 논문에서 제안하는 알고리즘이 적용된 모듈은 회색 블록으로 표현하였다. Current CU의

최적의 모드가 Affine merge 모드이면서 동시에 Affine AMVR 모드가 정수형이 아닌 경우 정수 단위 Affine ME (AME)를 조기종료한다 추가로 Parent CU의 4-parameter affine 모드가 AMVP 모드와 6-parameter affine 모드보다 RD-cost가 낮은 경우 6-parameter AME를 조기종료한다.

#### 4. 실험 결과

제안방법은 Table 1과 같이 JVET Common Test Condition (CTC)를 기반으로 수행하였으며, 이때 Table 2는 비교실험이 수행된 실험환경을 보여준다. 제안방법 적용에 따른 부호화기 복잡도 절감효과를 측정하기 위해 식 (4)에 따라 JVET CTC에서 정의된 4개의 양자화 파라미터에서 측정한 복잡도 감소율(Time saving, TS)의 평균을 측정하였다.

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

여기서  $AMT_{org}$ 와  $AMT_{fast}$ 는 VTM 17.0의 AMT와 본 논문에서 제안하는 고속화 알고리즘이 적용된 AMT를 의미한다. AMT의 계산 복잡도는 QP값에 따라 변동될 수 있으므로 각 테스트 영상

Table 1. Encoding parameters used in the experiments

Encoding parameters	Options
QP	22, 27, 32, 37
Num. of referenceframes	2
Num. of encoding frames	33
Search range	384
Coding configuration	Random access
Use hadamard	Enable
GOP size	32

Table 2. Experimental environments.

Experimental environment	Options
Processor	Intel(R) Xeon(R) E5-2697 v3
RAM	256GB
Oprating system	64-bit Windows 10
C++ compiler	Microsoft Visual C++ 2019

Table 3. Coding loss and complexity of the proposed method on top of VTM 17.0

Class	Sequences	Proposed Method				
		BDBR-Y	BDBR-U	BDBR-V	TSAMT	TSTET
UHD	Tango2	0.28%	0.08%	0.06%	87%	98%
	FoodMarket4	-0.06%	0.25%	0.14%	86%	97%
	Campfire	0.10%	0.00%	0.02%	86%	103%
	CatRobot	0.45%	0.39%	0.65%	83%	95%
	DaylightRoad2	0.40%	0.33%	0.49%	84%	97%
	ParkRunning3	0.19%	0.04%	0.12%	90%	101%
FHD	MarketPlace	0.27%	1.56%	0.67%	84%	96%
	RitualDance	0.22%	0.78%	0.64%	83%	97%
	Cactus	0.11%	0.61%	0.38%	81%	97%
	BasketballDrive	-0.05%	-0.83%	0.03%	81%	97%
	BQTerrace	-0.04%	-0.26%	0.57%	79%	97%
Average		0.17%	0.27%	0.34%	84%	98%

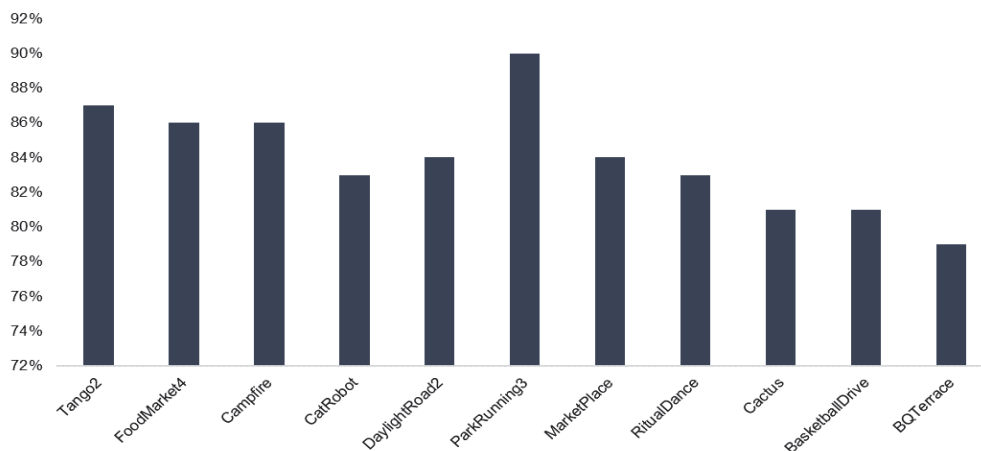


Fig. 8 Complexity reduction between the proposed method and VTM 17.0

의 TS는 AMT의 평균과 VTM 17.0 대비 총 인코딩 시간 ( $T_{enc}$ )으로 표현하였다. AMT를 계산하는 것과 유사하게, 평균 TS는 다른 QP값에 해당하는 4개의  $T_{enc}$  결과로부터 계산되었다. Table 3은 제안 방법과 기존 방법의 복잡성 감소를 보여준다. 추가로 고속화에 따른 코딩 손실을 평가하기 위해 BDBR (Bjontegaard Delta Bit Rate)을 측정하였다. 일반적으로 BDBR이 1% 증가하면 0.05dB의 BD-PSNR 감소에 해당하며, 여기서 BDBR의 증가는 코딩 손실을 나타낸다. 실험 결과 제안 방법은 VTM 대비 0.17%의 손실로 AME의 시간을 16%까지 감소하였다. 특히, CatRobot의 경우 0.45%의 손실을 나타내는 반면 부호화기의 복잡도가 5% 개선된 것을 보여주게 된다.

그리고 FoodMarket4, BasketballDrive 와 BQTerrace의 경우에는 BDBR-Y의 코딩 계인을 보유함과 동시에 부호화기의 복잡도가 감소하였다.

## 5. 결론

VVC는 인코딩 복잡도를 크게 증가시키는 반면,

평행 운동 모델의 한계를 극복하기 위하여 Affine 예측방법을 새롭게 채택하였다. 본 논문에서는 최적의 CU 모드 결정을 위한 RD 최적화 과정에서 얻을 수 있는 정보를 기반으로 Affine 고속화 알고리즘을 제안하였다. 제안 방법은 JVET CTC에서 평가되었으며, VTM 17.0과 비교하여 성능을 측정하였다. 계산 복잡도를 비교하기 위하여 Affine 예측에 소요되는 인코딩 시간을 측정하였으며, 실험 결과 제안 방법은 VTM 대비 0.17%의 손실로 AME의 시간을 16%까지 감소하였다.

## 사 사

이 논문은 동아대학교 교내연구비 지원에 의하여 연구되었음

## 참고문헌

- [1] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, Overview of the High Efficiency

- Video Coding (HEVC) Standard, IEEE Transactions on Circuits and Systems for Video Technology, 22, 12, pp.1649-1668, (2012).
- [2] B. Bross, Y.K. Wang, Y. Ye, S. Liu, J. Chen, and G.J. Sullivan, et al., Overview of the Versatile Video Coding (VVC) Standard and Its Applications, IEEE Transactions on Circuits and Systems for Video Technology, 31, 10, pp.3736-3764 (2021).
- [3] Chien, W.; Boyce, J. JVET AHG report: Tool reporting procedure (AHG13), JVET-S0013, JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Teleconference (2020).
- [4] K. Zhang, Y.-W. Chen, L. Zhang, W.-J. Chien, and M. Karczewicz, An improved framework of affine motion compensation in video coding, IEEE Trans. Image Process., 28, 3, pp.1456-1469, (2019).
- [5] K. Zhang, L. Zhang, H. Liu, Y. Wang, P. Zhao, and D. Hong, CE4: Affine Prediction with 4-4 Sub-blocks for Chroma Components (Test 4.1.16), document JVET-L0265, (2018).
- [6] H. Huang, W.-J. Chien, Y. Han, Y. Zhang, and M. Karczewicz, CE4.1.6: Simplification of Affine AMVP Candidate List Construction, document JVET-L0271, (2018).

---

(접수: 2022.07.29. 수정: 2022.08.09. 게재확정: 2022.08.10.)