

# 적응적 탐색 영역을 이용한 HEVC 고속 움직임 탐색 방법

이 호 영\*, 심 혁 재\*, 박 영 현\*, 전 병 우°

## Fast Motion Estimation Using Adaptive Search Range for HEVC

Hoyoung Lee\*, Huik Jae Shim\*,  
Younghyeon Park\*, Byeungwoo Jeon°

### 요 약

본 논문에서는 HEVC 부호화의 연산 복잡도를 줄이기 위한 고속 움직임 탐색 방법을 제안한다. 고속 움직임 탐색을 위한 종래의 방법은 현재 영상과 참조 영상 간의 거리에 따라 탐색 영역의 크기를 조절하는 반면, 제안 방법은 예측 단위 간의 움직임 벡터 차의 유사성을 이용하여 예측 단위에 따라 움직임 벡터 탐색 영역을 적응적으로 조절한다. 실험을 통해 제안 방법은 random access 실험 조건에서 움직임 예측의 연산 시간을 약 10.7% 감소시키는 반면, 약 0.1%의 낮은 부호화 성능 하락을 가지는 것을 확인하였다.

**Key Words** : HEVC, Fast Motion estimation,  
Adaptive Search Range

### ABSTRACT

This paper proposes a fast motion estimation method which can reduce the computational complexity of HEVC encoding process. While the previous method determines its search range based on a distance between a current and a reference pictures to accelerate the time-consuming motion estimation, the proposed method adaptively sets the search range according to motion vector difference between prediction units. Experimental results show

that the proposed method achieves about 10.7% of reduction in processing time of motion estimation under the random access configuration whereas its coding efficiency loss is less than 0.1%.

### I. 서 론

Ultra High Definition (UHD)과 같은 고해상도 및 고화질 비디오 서비스에 대한 요구가 커짐에 따라, 종래의 비디오 부호화 기술보다 더 높은 부호화 성능을 가지는 새로운 표준 기술에 대한 필요성이 대두되었다. High Efficiency Video Coding (HEVC)은 이러한 필요성을 충족시키기 위해 가장 최근에 표준화가 완료된 비디오 부호화 기술이다 [1].

HEVC는 하나의 슬라이스를 Coding Tree Unit (CTU)으로 나누고, 각각의 CTU를 쿼드 트리 형태로 재귀적으로 분할하는 형태를 가진다. 이때 분할되는 단위를 코딩 단위 (Coding Unit: CU)라고 하며, 각 CU는 다시  $2N \times 2N$ ,  $N \times 2N$ ,  $2N \times N$ ,  $N \times N$ , 비대칭 분할 구조 (Asymmetric Motion Partitions: AMP)의 다양한 예측 단위 (Prediction Unit: PU)로 분할된다. HEVC는 가능한 모든 분할 구조에 대해 움직임 예측 및 보상을 수행하여 최적의 분할 구조를 찾는다. 이러한 유연한 분할 구조와 같은 좀 더 정교한 기술은 HEVC가 종래의 H.264/MPEG-4 AVC에 비해 동일한 주관적 화질에서 약 50%의 압축 성능 향상을 달성하는데 중요한 역할을 했으나, 부호화기의 연산 복잡도 또한 크게 증가시켰다. 특히, 움직임 예측은 최적의 움직임 벡터를 찾기 위해 다양한 CU/PU 구조 및 복수개의 참조 영상에 대해 실행되기 때문에 부호화기에서 가장 높은 연산 복잡도를 차지하며, 이러한 문제를 해결하기 위해 다양한 연구들이 진행되었다 [2,3].

본 논문에서는 움직임 예측의 연산 복잡도를 감소시키기 위해 PU 간의 움직임 벡터 차 간의 유사성을 이용하여, PU에 따라 움직임 탐색 영역을 적응적으로 적용하는 방법을 제안한다. 이를 위하여 제안 방법은 CU에서 분할되는 PU들 간의 유사성을 확률적으로 계산하고, 이를 바탕으로 각 PU에 대한 움직임 탐색 영역의 크기를 조절한다.

\* 이 논문은 2011년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2011-001-7578).

• First Author : Sungkyunkwan University Department of Electronic and Electrical Engineering, hoijing@skku.edu, 정희원

° Corresponding Author : Sungkyunkwan University Department of Electronic and Electrical Engineering, bjeon@skku.edu, 정희원

\* Sungkyunkwan University Department of Electronic and Electrical Engineering, {waitnual, neversky}@skku.edu, 중신희원, 학생회원  
논문번호 : KICS2014-03-075, Received March 5, 2014; Revised March 28, 2014; Accepted March 28, 2014

## II. 제안 방법

HEVC의 움직임 예측은 정화소 단위 움직임 탐색과 부화소 단위 움직임 탐색으로 구성된다. 먼저, 예측 움직임 벡터 (Predicted Motion vector: PMV)를 결정하고, PMV를 시작점으로 하여 사용자가 지정한 탐색 영역 내에서 정화소 단위 움직임 탐색을 수행하여 최적 지점을 찾는다. HEVC는 정화소 단위 움직임 탐색에서 찾은 최적 지점을 중심으로 다시 부화소 단위 움직임 탐색을 수행하여 최적의 움직임 벡터 (Motion vector: MV)를 찾는다. 이러한 움직임 예측을 통해 결정된 움직임 벡터와 PMV와의 차를 계산한 움직임 벡터 차 (Motion vector difference: MVD)는 엔트로피 부호화 된다. 이러한 정화소 단위 움직임 탐색은 전체 부호화기의 연산 복잡도 중에서 약 40%의 높은 연산 복잡도를 차지하고 있다 [4].

정화소 단위 움직임 탐색의 연산 복잡도를 줄이기 위해, 현재 영상과 참조 영상간의 거리에 따라 탐색 영역의 크기를 가변적으로 조절하는 방법이 개발되었다[5]. 이와 달리 제안 방법은 현재 CU가 특정 물체 (Object) 내에 존재할 경우, CU 내의 모든 PU들의 MV가 유사할 가능성이 높음을 가정하여,  $2N \times 2N$  PU의 MVD를 바탕으로 그 외 PU의 탐색 영역을 조절한다. 표 1은  $2N \times 2N$  PU의 MVD와 그 외 PU의 MVD 간의 차이의 확률을 나타낸다. 표 1에서  $POC_{Dist}$ 는 현재 영상과 참조 영상 간의 시간 거리를 나타내고, SMP (Symmetric Motion Partitions)는  $2N \times N$ ,  $N \times 2N$ 과 같은 대칭적 분할 구조를 나타낸다.  $Dif$ 는  $2N \times 2N$  PU의 MVD와 그 외 PU의 MVD 간의 차이를 나타내며 다음과 같이 계산한다.

표 1.  $2N \times 2N$  PU와 그 외 PU의 MVD간의 차의 발생 확률  
Table 1. Probability of MVD difference between  $2N \times 2N$  PU and the other PUs

$POC_{Dist}$	PU type	$Prob(x)$			
		$x:Dif=0$	$x:Dif \leq 4$	$x:Dif \leq 8$	$x:Dif \leq 16$
1	SMP	0.71	0.83	0.86	0.90
	AMP	0.46	0.75	0.83	0.88
2	SMP	0.59	0.8	0.82	0.86
	AMP	0.40	0.72	0.78	0.82
3	SMP	0.79	0.94	0.96	0.97
	AMP	0.39	0.69	0.77	0.81
4	SMP	0.58	0.84	0.86	0.89
	AMP	0.34	0.69	0.77	0.79
Avg	SMP	0.67	0.85	0.87	0.90
	AMP	0.39	0.71	0.79	0.82

SMP (Symmetric Motion Partitions);  
AMP (Asymmetric Motion Partitions)

$$Dif = |MVD_{Max}^{2N \times 2N} - MVD_{Max}^{nN \times nN}|$$

$$where MVD_{Max} = Max(|MVD_x|, |MVD_y|) \quad (1)$$

여기에서  $MVD_x$ 와  $MVD_y$ 는 각각  $x$ 축과  $y$ 축 방향의 MVD 값을 나타내며,  $Max(\cdot)$ 는  $MVD_x$ 와  $MVD_y$  중 더 큰 값을 나타낸다. 표 1에서 보는 바와 같이 SMP PU들은  $2N \times 2N$  PU의 MVD와의 차이가  $\pm 4$  이내일 확률이 약 0.8 이고, AMP PU들은  $2N \times 2N$  PU의 MVD와의 차이가  $\pm 8$  이내일 확률이 약 0.8 임을 확인할 수 있었다. 제안 방법은 현재 CU에서  $2N \times 2N$  PU의 MVD를 기준으로, 그 외 PU의 탐색 영역을 재설정한다.

그림 1은 제안 방법의 블록 다이어그램을 나타낸다. 여기에서 보는 바와 같이 제안 방법은  $2N \times 2N$  PU에 대해서는 기본 탐색 영역인  $\pm 64$ 를 사용하며, 그 외 PU에 대해서는 수식 (2)와 같이 탐색 영역을 재설정하여 사용한다.

$$SR = \begin{cases} MVD_{Max}^{2N \times 2N} + 4 & \text{for SMP PUs} \\ MVD_{Max}^{2N \times 2N} + 8 & \text{for AMP PUs} \end{cases} \quad (2)$$

여기에서  $SR$ 은 새로운 움직임 탐색 영역을 나타낸다. 식 (2)에서 보는 바와 같이 제안 방법은  $2N \times 2N$  PU의  $MVD_{Max}$ 에 따라, 그 외 PU의 탐색 영역을 적응적으로 설정한다.

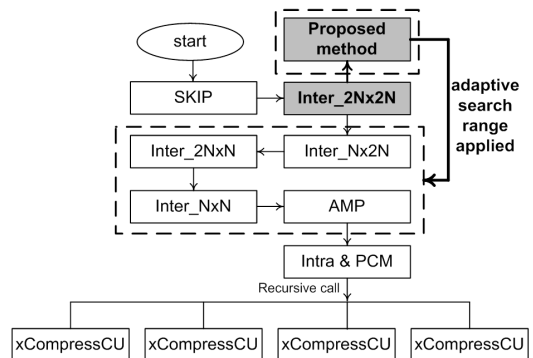


그림 1. 제안 방법의 블록 다이어그램  
Fig. 1. Block diagram of the proposed method

## III. 실험 결과

제안 방법의 성능을 검증하기 위해, HEVC 참조 소프트웨어인 HM11.0[5]에 제안 방법을 구현하고, 부호화 효율 및 움직임 예측의 연산 시간 감소를 측정하였다. random access main scenario 실험 조건에서,

8개의 영상을 사용하여 실험을 수행하였다. 제안 방법의 성능을 평가하기 위해, 종래의 방법<sup>[5]</sup>과 그 성능을 비교하였다. 부호화 효율의 성능 평가를 위해 HEVC 참조 소프트웨어 HM11.0을 기준으로 제안 방법과 종래의 방법<sup>[5]</sup>에 대한 Bjøntegaard delta bit rate (BDBR)<sup>[6]</sup>을 측정하였고, 식 (3)과 같이 움직임 예측의 연산 시간 감소를 측정하였다.

$$TS(\%) = \frac{PT(Anchor) - PT(Proposed)}{PT(Anchor)} \times 100 \quad (3)$$

여기에서 PT (Processing Time)는 움직임 예측의 연산 시간을 나타내고, TS (Time saving)는 제안 방법의 연산 시간 감소량을 나타낸다. BDBR은 그 값이 양수일 경우 제안방법의 부호화 효율이 하락하였음을 의미하고, TS는 값이 음수일 경우 제안방법의 연산 시간이 증가하였음을 나타낸다.

표 2는 종래의 방법과 제안 방법의 성능을 비교한 결과이다. 여기에서 보는 바와 같이 종래의 방법이 평균적으로 약 8.3% 연산 시간 감소 및 약 0.2%의 부호화 효율 하락을 가지는 것과 비교하여, 제안 방법은 종래의 방법에 비해 약 2.4%의 연산 시간이 더 감소하였고, 부호화 효율 또한 0.1% 더 뛰어난 성능을 보임을 확인할 수 있다. 특히, 제안 방법은 Traffic, Cactus, RaceHorses와 같이 움직임이 빠른 영역이 존재하는 영상에서 종래의 방법에 비해 부호화 효율 및

표 2. 제안 방법과 종래 방법의 성능 비교  
Table 2. Performance comparison between the proposed method and conventional method

method	size	Sequence	BDBR (%)	TS (%)
conventional method <sup>[5]</sup>	2560×1600	Traffic	0.4	1.7
		PeopleOnStreet	0.0	12.2
	1920×1080	Kimono	0.3	11.3
		Cactus	0.6	7.2
	832×480	BQMall	0.0	5.2
		RaceHorses	0.4	15.9
	240×416	BasketballPass	0.1	8.6
		BlowingBubbles	0.0	4.1
	Average	0.2	8.3	
Proposed	2560×1600	Traffic	0.1	3.5
		PeopleOnStreet	0.1	14.6
	1920×1080	Kimono	0.1	14.9
		Cactus	0.1	8.6
	832×480	BQMall	0.1	7.9
		RaceHorses	0.2	17.1
	240×416	BasketballPass	0.0	11.8
		BlowingBubbles	0.1	6.7
	Average	0.1	10.7	

연산 시간 감소 측면에서 뛰어난 성능을 보이는 것을 확인할 수 있다.

#### IV. 결 론

본 논문에서는 적응적 움직임 탐색 영역을 이용한 HEVC 고속 움직임 방법을 제안하였다. 제안 방법은 각 CU 별로 움직임 탐색 영역을 적응적으로 설정함으로써 움직임 예측의 연산 시간을 감소시킨다. 실험을 통해 제안 방법은 random access 실험 조건에서 약 10.7%의 움직임 예측의 연산 시간 감소를 달성하였으며, 부호화 효율 하락은 약 0.1%로 미미한 것을 확인하였다. 이것은 종래의 방법과 비교하여 더욱 개선된 성능이다.

#### References

- [1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [2] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 558-564, Feb. 2014.
- [3] D. Yoon and Y. Ho, "Fast mode decision method for HEVC in depth video," *J. KICS*, vol. 37, no. 1, pp. 51-56, Jan. 2012.
- [4] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation Analysis," *IEEE Trans. on Circuits and Syst. for Video Tech.*, vol. 22, no. 12, pp. 1685-1696, Dec. 2012.
- [5] High Efficiency Video Coding Test Model Software 11.0, Available : [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware)
- [6] G. Bjøntegaard, *Calculation of average PSNR differences between RD-Curves*, ITU-T SG16 Q.6 document VCEG-M33, Switzerland, Apr. 2001.