

# HEVC 부호기에서의 고속 CU 결정 방법

김대연, 이용기, 김형덕

픽스트리

dykim@pixtree.com, yglee@pixtree.com, donald@pixtree.com

## Fast CU Decision Method for HEVC Encoder

Kim, Dae-Yeon, Lee, Yung-Gi, Kim, Hyeong-Duck

PIXTREE

### 요약

현재 표준화가 진행 중인 HEVC (High-efficiency video coding) 는 기존의 동영상 표준과 마찬가지로 여러 기술들이 혼합된 하이브리드 영상 부호화 프레임 워크 구조를 따르고 있다. 특히, 다양한 크기의 부호화 단위 (Coding Unit : CU), 예측 단위 (Prediction Unit : PU), 변환 단위 (Transform Unit : TU) 의 사용으로 인해 HD 이상의 영상에 대하여 기존의 H.264/AVC 보다 약 40%의 압축률 향상을 보이고 있다. 하지만 그로 인하여 부호화기 복잡도가 약 3 배 이상 증가하는 것으로 나타났으며 이는 실시간 부호화가 요구되는 분야에서 큰 문제가 될 것이다. 본 논문은 HEVC 부호화기 복잡도를 낮추기 위하여 최적의 CU 를 결정하는 과정 중 초기에 CU 를 결정하는 고속 CU 결정 방법을 소개한다. 실험 결과, 제안된 방법은 HM 과 비교하여 PSNR (Peak Signal to Noise Ratio) 의 손실이 거의 없이 최대 약 58%의 부호화 시간을 절약하였다.

## 1. 서론

HEVC Test Model (HM) [1]은 그 모태가 되는 Test Model under Consideration (TMuC) 에서 부호기 및 복호기에서 계산 복잡도가 낮으면서 압축 성능이 우수한 몇 개의 중요한 coding tool 들을 포함하는 것을 시작으로 coding tool 을 변형시켜 성능을 향상 시키거나 새로운 기술들이 포함되어 현재 HM 3.1 까지 배포되었다. HM 에서는 두가지 실험 설정 (High efficiency, low complexity)이 정의되어 있다. High efficiency 설정은 계산 복잡도 보다는 압축 성능에 초점을 맞춘 설정들의 조합이며, low complexity는 저복잡도에 초점을 맞춘 설정들의 조합이다. H.264/AVC [2]와 비교하면, high efficiency는 main profile, low complexity 는 baseline profile 이라고 생각해도 무방할 것이다. 각 세팅에서 사용되는 coding tool 들의 조합은 다음 표 1 과 같다.[3] 표와 같이 Angular prediction, AMVP (Advanced Motion Vector Prediction), DCT-IF (DCT-based Interpolation Filter), ALF (Adaptive Loop Filter), RQT (Residual Quadtree) 등의 coding tool 들과 다양한 크기의 부호화 단위, 예측 단위, 변환 단위의 사용과 같은 새로운 부호화 구조로 인하여 high efficiency 설정에서 H.264/AVC 대비 약 35 %의 압축률 향상을 보이는 것으로 보고되고 있다. 하지만 그로 인하여 부호화기 복잡도가 크게 증가되었고 H.264/AVC 대비 약 3 배 증가하는 것으로 나타났다.[4] 따라서 실시간 부호화가 필요한 다양한 응용 분야에서 HEVC 가 사용되기 위해서는 압축 성능을 유지하면서 부호화 시간을 줄이는 연구가 필요하다. 본 논문에서는 부호화기 복잡도를 낮추기 위하여 고속 CU 결정 방법을 제안하였다.

다음 장에서는 HEVC 의 CU 에 대해 설명하고 3장에서 제안된 고속 CU 결정 방법을 설명한다. 그리고 제안된 방법에 대한 실험 결과를 4장에 보여주고 5장에 결론을 맺는다.

표 1. HM 실험 환경

High Efficiency	Low Complexity
Coding Unit tree structure (8x8 up to 64x64 luma samples)	
Prediction Units (2Nx2N, 2NxN, Nx2N, NxN)	
Transform unit tree structure (3 level max.)	Transform unit tree structure (2 level max.)
Transform block size of 4x4 to 32x32 samples (always square)	
Angular Intra Prediction (34 directions max.)	
DCT-based interpolation filter for luma samples	
DCT-based interpolation filter for chroma samples	
AMVP (Advanced motion vector prediction)	
CABAC	CAVLC
Internal bit-depth increase	X
X	Transform precision extension
Deblocking filter	
Adaptive loop filter	X

## 2. CU 와 HM 의 CU 결정 방법

### 가. LCU 와 CU

LCU (Largest CU) 는 부호화 기본 단위로써 H.264/AVC 의 MB (Macroblock)와 동일한 개념이다. 하나의 LCU는 NxN 크기의 휘도 샘플과 그에 대응하는 두개의 N/2xN/2 색차 샘플로 구성된다. (4:2:0 포맷으로 가정). 현재 HM 에서는 LCU는 64x64를 사용하기 때문에 본 논문에서는 LCU의 크기를 64x64로 가정하고 설명하도록 하겠다.[1]

CU 는 화면내 예측을 사용했는지 혹은 화면간 예측을 사용했는지를 구분하기 위한 기본 단위이다. 항상 정사각형 모양을 가지며, LCU 크기 이하의 파티션들이 사용되며, 항상 8x8 보다는 크다. 하나의 LCU

표 2 최대 CU depth가 4 일 경우, 각 CU 에 대한 depth index

CU	CU depth
64x64 CU (LCU)	0
32x32 CU	1
16x16 CU	2
8x8 CU	3

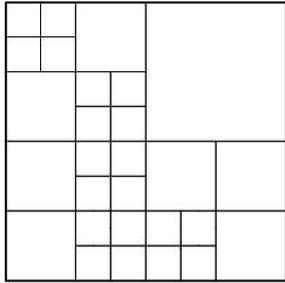


그림 1. CU 구조의 예

내에는 하나 이상의 다양한 크기의 CU 들이 존재할 수 있으며, LCU 를 시작으로 4진 트리(Quadtree) 구조로 재귀적으로 4등분된다. 그림 1은 하나의 LCU 가 다수의/다양한 크기의 CU 로 부호화된 것을 보여주는 예이다.

현재 CU 의 크기를 나타내기 위한 값으로 CU 깊이 (CU depth) 를 사용한다. 이 값은 LCU 를 기준으로 현재 CU 가 4진 트리 형태로 얼마나 작아졌는지를 나타내는 값이다. LCU 의 depth index는 0 이며, CU 의 크기가 LCU 보다 작아질수록 depth index 는 1씩 증가한다. HM 에서는 최대 CU depth 를 4 로 설정하며, 이 경우 사용될 수 있는 CU 와 이에 대응되는 depth index 는 표 2와 같다. 색차 샘플에 대한 CU 는 휘도 샘플에서 결정된 CU 크기에 비례한 크기를 사용한다. 예를 들어, 4:2:0 포맷에서 휘도의 CU 가 16x16 이면 색차의 CU 는 8x8 이 된다.

### 나. HM 에서의 CU 결정 방법

HM 부호기는 기본적으로 율-왜곡 최적화 (Rate-Distortion Optimization) [5] 기반으로 CU 를 결정한다. 그림 2는 HM 부호기에서의 CU 결정 과정을 그린 흐름도이다. 그림 2와 같이 HM 은 64x64 크기의 LCU 를 시작으로 현재 CU 에서 사용 가능한 PU 들과 TU 를 이용하여 율-왜곡량 (RD cost) 이 가장 작은 PU 와 TU 의 조합을 선택하고 이때의 RD cost 와 현재 CU 를 4등분하였을 때의 4개의 sub CU 의 RD cost 합과 비교하여 작은 RD cost 값을 갖는 CU 를 선택한다. 이 과정이 재귀적으로 CU depth 가 증가하면서 동일하게 반복된다. 따라서 항상 LCU 에서부터 최소 크기 CU 까지의 모든 CU, PU, TU 에 대한 모든 조합 중 RD cost 가 최소인 조합을 선택할 수 있기 때문에 최대의 압축 효율을 나타낼 수 있다. 하지만 각 CU depth 는 최대 7가지의 PU 모드 (Merge, Inter2Nx2N, Inter2NxN, InterNx2N, InterNxN, Intra2Nx2N, IntraNxN) 와 최대 3가지의 TU (2Nx2N, NxN, N/2xN/x)에 대하여 실제로 부호화를 수행해야 하기 때문에 계산 복잡도가 상당히 높다.

### 3. 제안하는 고속 CU 결정 방법

일반적으로 하나의 LCU 가 부호화될 때, 텍스처가 복잡하거나 움직임이 많은 영역일 경우에는 작은 CU 와 PU 로 선택된다. 이런 경우에는 보다 다수의 작은 CU 와 PU 가 사용되었을 때 압축 효율이 높기 때문이다. 반대로 텍스처가 완만하거나 움직임이 거의 없는 영역에서는 대부분 큰 CU 와 PU 가 사용될 때 압축 효율이 높을 것이다. 이러한 가정을 바탕으로 CU depth 가 d 일 때가 d+1 일 때보다 압축 효율이 더 우수하다고 판단된다면 d+2 이상의 CU depth 를 수행하지 않아도 압축 효율의 손실을 최소화하면서 부호화 속도를 향상시킬 수 있을 것이다. 제안된 방법은 다음과 같은 과정으로 CU 를 결정한다.

- Step 1. i 번째 LCU 부호화 시작 (CU depth d = 0)
- Step 2. CU depth d 에 대한 PU, TU 결정
  - if  $3 < d < 0$ , 후보 PU 모드 = {BestPU<sub>d</sub>, Skip, Merge}
  - otherwise, 후보 PU 모드 = {모든 PU 모드}
  - RDCost<sub>d</sub> = 최소 RD cost 저장
  - if d = 3, Step 5 로 이동
- Step 3. 4개의 CU depth d + 1 에 대하여 각각 PU, TU 결정 (d+2 이상의 CU depth 는 이용하지 않음)
  - BestPU<sub>d+1</sub> = 최적의 PU 모드 저장
  - RDCost<sub>d+1</sub> = 4개 CU의 최소 RD cost 의 합 저장
- Step 4. RDCost<sub>d</sub> 와 RDCost<sub>d+1</sub> 비교
  - if RDCost<sub>d</sub> < RDCost<sub>d+1</sub>, Step 6 으로 이동
  - otherwise, Step 5 로 이동
- Step 5. CU depth 증가 (d = d+1)
  - if d > 3, Step 6 으로 이동
  - otherwise, Step 2 로 이동
- Step 6. 현재까지 정보를 이용하여 i 번째 LCU 를 엔트로피 부호화, 다음 i+1 번째 LCU 로 이동

다음 그림 3은 제안된 고속 CU 결정 과정을 나타낸 그림이다. 그림 2의 HM 에서의 CU 결정 과정과 비교하여 32x32 CU 에 대한 PU 와 TU 결정 과정, 16x16 CU 에 대한 PU 와 TU 결정 과정이 제안된 방법에는 한 번 더 수행이 된다. 이것은 최악의 경우 계산 복잡도를 오히려 증가시킬 수 있기 때문에 앞서서 결정된 32x32 또는 16x16 CU 의 PU 모드 정보를 이후에 수행되는 동일 CU 에서 이용하였다. 이 때, MV (Motion Vector) 정보나 TU 정보도 이용할 수 있으나 많은 정보를 이용할수록 압축 효율 감소가 크다. 그 이유는 이 정보들이 CU 를 sub CU 로 나누지 않고 현재 CU 에서의 최선의 PU, TU, MV 정보들이기 때문에 만약 이전의 CU 가 4개의 sub CU 로 나뉘져 부호화되었을 경우 현재 CU 의 주변 정보들이 바뀌면서 이 정보들이 최선이 아니게 될 수 있기 때문이다. 따라서 제안된 방법에서는 PU 모드 정보만을 이용하여 부호화 효율의 손실을 최소화하였다. 또한 부호화 효율의 손실을 더욱 최소화하기 위하여 이전에 결정된 PU 모드와 통계적으로 가장 많이 최적의 모드로 선택되는 Skip 모드와 Merge 모드를 후보 모드로 사용하여 율-왜곡 최적화를 통해 최적의 모드를 결정하였다.

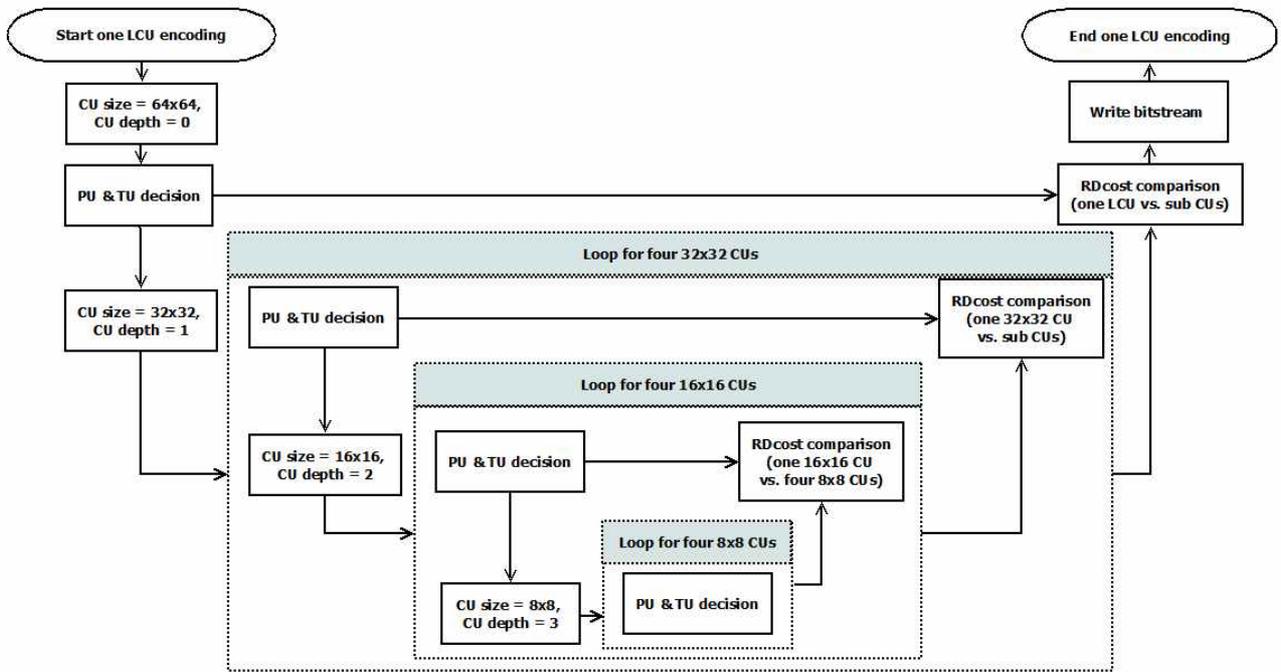


그림 2. HM 에서의 CU 결정 과정에 대한 흐름도

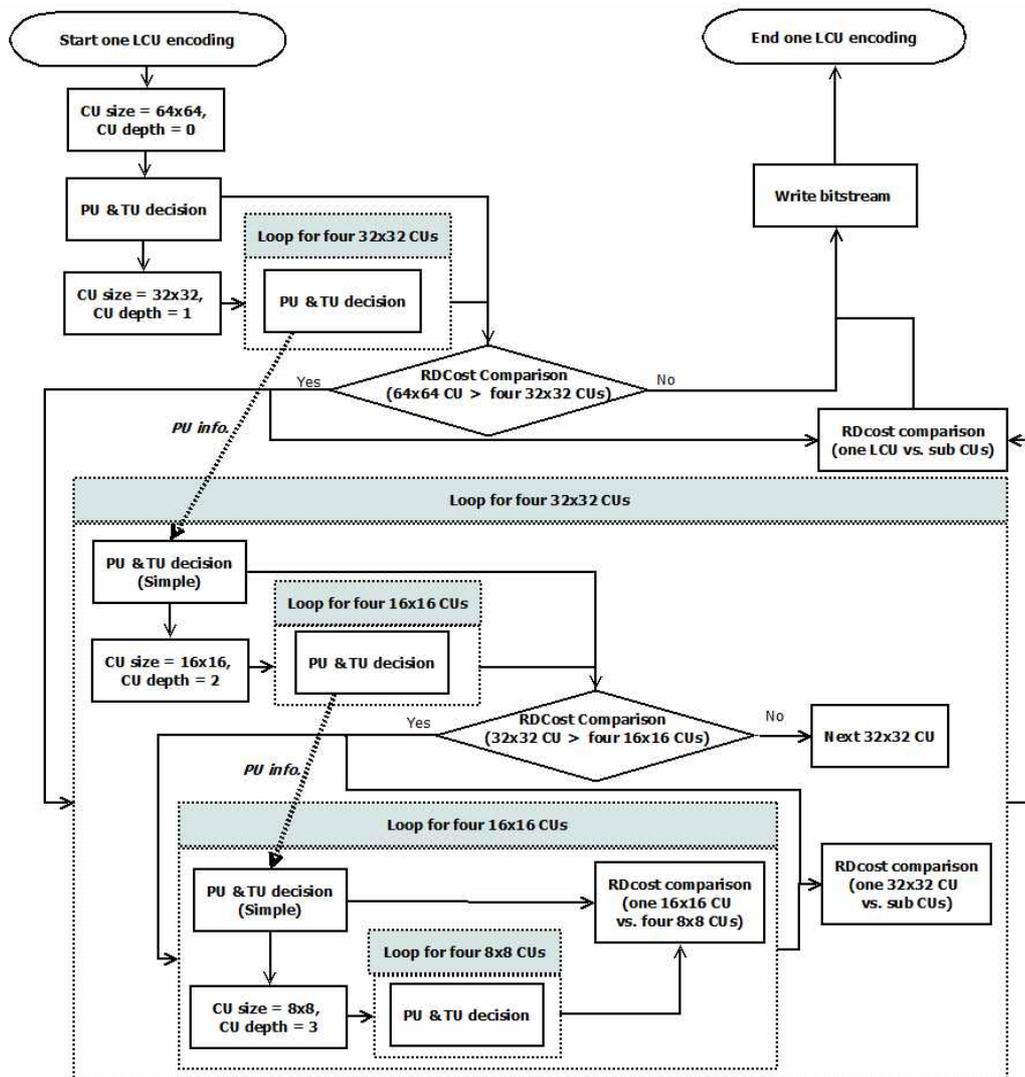


그림 3. 제안된 CU 결정 방법에 대한 흐름도

표 3. 실험에 사용된 영상과 실험 조건

Sequences	Resolution	Experimental conditions
Kimono (24Hz)	1920 x 1080	Random access (Hierarchical B, GOP size = 8) QP={22, 27, 32, 37} [3] 참고
ParkScene (24Hz)		
Cactus (50Hz)		
BQTerrace (60Hz)		
BasketballDrive (50Hz)		

#### 4. 실험 결과

제안된 방법의 성능을 평가하기 위하여 HEVC의 참조 소프트웨어 HM 2.1에 제안된 방법을 구현하였다[6]. 실험은 Intel Core i7-2600 3.4 GHz, 8Gbyte RAM 사양의 PC에서 수행되었고 1080p 해상도의 영상에 대하여 각각 100 프레임을 부호화하였다. 자세한 실험 조건은 표 3에 나타내었다.

표 4는 실험 결과를 나타낸 표이다. 표 4에서  $\Delta$ BD-RATE는 [7]에 정의되어 있으며, 울-왜곡 곡선 상에서의 평균 비트율 증가량을 나타낸다. 부호화 시간 절약을  $\Delta T$ 는 다음 식 (1)과 같이 계산하였다.

$$\Delta T(\%) = \frac{T_{HM} - T_{proposed}}{T_{HM}} \times 100 \quad (1)$$

식 (1)에서  $T_{HM}$ 은 HM의 전체 부호화 시간이며,  $T_{proposed}$ 는 제안된 방법에 의한 전체 부호화 시간이다.

표 4로부터 제안된 방법은 HM과 비교하여 high efficiency에서는 평균적으로 0.9%의 비트율이 증가하지만 약 39.5%의 부호화 시간을 절약하였으며, low complexity에서는 평균적으로 0.7%의 비트율이 증가하지만, 약 49.2%의 부호화 시간을 절약하였다. Low complexity에서는 복잡도가 높은 ALF와 같은 coding tool이 사용되지 않아 전체 부호화 과정에서 CU 결정 과정이 차지하는 비율이 높다. 따라서 high efficiency보다 low complexity 설정에서 제안된 방법의 속도 향상률이 약 10% 더 높은 결과를 보였다.

#### 5. 결론

본 논문은 HEVC 부호화기 복잡도를 낮추기 위하여 최적의 CU를 결정하는 과정 중 조기에 CU를 결정하는 고속 CU 결정 방법을 제안하였다. 실험 결과, 제안된 방법은 HM과 비교하여 압축 성능의 손실이 거의 없이 최대 약 58%의 부호화 속도 향상을 나타내었다. 차후 PU 모드 정보뿐만 아니라 MV, TU 정보를 효과적으로 재이용하는 방법을 연구한다면 제안된 방법은 실시간 부호화를 요구하는 분야에서 유용하게 사용될 수 있을 것으로 기대된다.

표 4. 실험 결과

Sequences	High efficiency		Low complexity	
	$\Delta$ BD-RATE (%)	$\Delta T$ (%)	$\Delta$ BD-RATE (%)	$\Delta T$ (%)
Kimono	0.5	45.1	0.4	58.0
ParkScene	0.8	37.5	0.5	50.4
Cactus	1.0	38.0	0.4	45.5
BasketballDrive	0.6	37.4	1.1	46.1
BQTerrace	1.4	39.7	1.3	46.1
<b>Average</b>	<b>0.9</b>	<b>39.5</b>	<b>0.7</b>	<b>49.2</b>

#### 참고 문헌

- [1] Thomas Wiegand, Woo-Jin Han, Benjamin Bross, Jens-Rainer Ohm, Gary J. Sullivan, "WD3: Working Draft 3 of High-Efficiency Video Coding", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5rd Meeting, Geneva, CH, 16-23 March, 2011
- [2] ITU-T Recommendation H.264 and ISO/IEC 14496-10, "Advanced video coding for generic audiovisual services", May 2003.
- [3] Frank Bossen, "Common test conditions and software reference configurations", Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th meeting, Daegu, KR, 20-28, January, 2011, Doc. JCTVC-D600.
- [4] S. Park, J. Park, B. Jeon, "Report on the evaluation of HM versus JM", Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th meeting, Daegu, KR, 20-28, January, 2011, Doc. JCTVC-D181.
- [5] Gary J. Sullivan and Thomas Wiegand, "Rate-Distortion Optimization for Video Compression", IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74-90, November, 1998.
- [6] <http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-2.1>
- [7] Gisle Bjontegaard, "Improvements of the BD-PSNR model", ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, July 2008, Doc. VCEG-A111.