

HEVC 코딩 트리 블록 분할 구조 고속 결정 방법

정순홍, 김희용
 한국전자통신연구원
 zeroone@etri.re.kr, hykim5@etri.re.kr

A Fast Partition Structure Decision Method in a Coding Tree Block of HEVC

Soon-heung Jung, Hui Yong Kim
 Electronics and Telecommunications Research Institute

요 약

본 논문에서는 HEVC 부호화시 코딩 트리 블록의 분할 구조를 고속 결정하는 방법을 제안한다. 코딩 트리 블록은 다양한 크기의 코딩 블록으로 구성되어 부호화 효율을 향상시키지만, 구성되는 코딩 블록을 결정하기 위한 과정에서 많은 계산량을 필요로 하게 되어 부호화 시간을 증가시킨다. 제안하는 방법에서는 부호화 과정에서 복원된 잔차신호와 코딩 트리 블록의 분할 구조의 상관성을 이용하여 코딩 트리 블록의 분할 구조를 고속으로 결정하는 방법을 제시한다. 실험 결과를 통해 제안된 방법이 HM16.0 에 비해 random-access configuration 에서 50.98%, low-delay configuration 에서 43.77%의 부호화 시간을 감소시키는 것을 확인하였다. 이때, BD-rate_{uv} 증가는 각각 2.42%와 2.35%로 부호화 효율에는 미치는 영향은 낮았다.

1. 서론

ITU-T/ISO/IEC JCT-VC(Joint Collaborative Team on Video Coding)에서는 AVC/H.264 대비 동일한 화질에서 2 배 이상의 부호화 효율을 제공하는 HEVC(High Efficiency Video Coding) 비디오 부호화 표준 기술을 개발하였다[1]. HEVC 기술에서는 높은 부호화 성능을 달성하기 위해서 많은 기술들이 추가되었고, 이로 인해 부호화기의 복잡도가 크게 증가하였다. 따라서 HEVC 기술이 시장에서 활용되기 위해서는 부호화기의 복잡도 감소를 통한 부호화 속도 개선이 꼭 필요하며, 다양한 연구들이 진행되었다[2] - [4].

전통적인 비디오 부호화기에서는 16x16(픽셀)로 이루어진 매크로블록을 기반으로 부호화가 이루어졌지만, HEVC 부호화기에서는 CTB(Coding Tree Block)를 기반으로 부호화가 이루어진다. CTB는 64x64, 32x32, 16x16의 크기를 가질 수 있으며, 일반적으로 64x64 를 많이 사용하고 있다. 이러한 CTB 를 기반으로 Quadtree Coding 이 이루어지게 되는데, CTB 내에서 Quadtree 형태로 나누어진 각 Block 에 해당하는 CB(Coding Block)를 기반으로 예측 및 변환 부호화가 이루어진다. CB 는 64x64, 32x32, 16x16, 8x8 의 값을 가질 수 있으며, CTB 내에서 Quadtree 구조를 가지면서 유효-왜곡 비용을 최소화시킬 수 있는 형태로 구성된다. 그림 1 은 64x64 CTB 가 Quadtree 구조를 가지면서 최소의 유효-왜곡 비용을 갖는 구조를 찾기 위해서 부호화가 필요한 CB 를 보여준다. HEVC test model (HM) 부호화기의 CTB 부호화 순서는 그림 2 와 같이 진행된다. 그림 3 은 이러한 부호화 과정

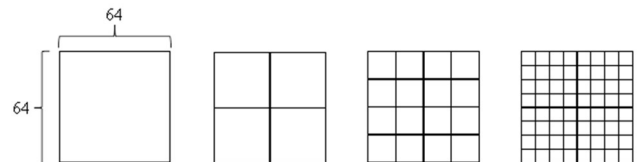


그림 1. 64x64 CTB 에서 부호화 되는 CB

을 모두 거친 후 결정된 64x64 CTB 에서 최소의 유효-왜곡 비용을 갖는 분할 구조의 한 예이며, 85 개의 CB 에 대해서 부호화 과정을 거쳐서 13 개의 CB 로 분할된 구조를 찾아낸다. 따라서, 64x64 CTB 의 분할구조를 예측하여 부호화할 CB 의 개수를 줄이면 부호화기의 복잡도를 상당량 감소시킬 수 있다.

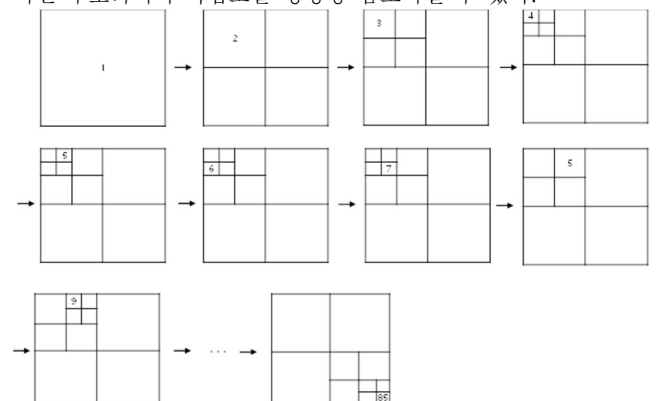


그림 2. HEVC test model (HM) 부호화기의 64x64 CTB 내 CB 의 부호화 순서

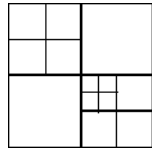


그림 3. 64x64 CTB 에서 최소 율-왜곡 비용을 갖는 분할 구조의 예

본 논문에서는 부호화 과정에서 복원된 잔차신호와 코딩 트리 블록 분할 구조의 상관성을 이용하여 HEVC 부호화시 코딩 트리 블록의 분할 구조를 고속으로 결정하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2 절에서는 부호화 과정에서 복원된 잔차신호와 코딩 트리 블록 분할 구조의 상관성 대해 살펴본 후, 3 절에서는 제안하는 코딩 트리 블록 분할 구조 고속 결정 방법을 설명하고, 4 절에서는 제안한 방법의 성능을 실험을 통해서 확인한다. 마지막으로 5 절에서 결론을 맺는다.

2. 복원된 잔차신호와 코딩 트리 블록의 분할 구조 상관성

코딩 트리 블록은 율-왜곡 비용을 최소화시키는 코딩 블록들로 분할되어 부호화된다. 그림 4 는 HM 인코더의 구조도를 나타낸다. 율-왜곡 비용 계산을 위해서는 입력신호와 예측 신호의 차이인 잔차신호와 이를 부호화하기 위해 필요한 비트량 정보가 사용된다[5]. 한편, 잔차신호는 변환/양자화 과정 후 엔트로피 부호화되는 부분과 역양자화/역변환 과정을 거치는 부분으로 나뉘어진다. 이때, 역양자화/역변환을 거친 복원된 잔차신호의 크기는 율-왜곡 비용 계산을 위해 사용되는 잔차신호와 이를 부호화하기 위한 비트량과 비례하게 된다.

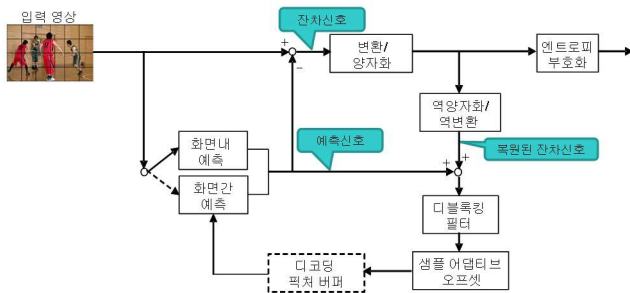


그림 4. HEVC test model (HM) 인코더의 구조도

그림 5 는 64x64 CTB 의 분할구조와 64x64 CB 의 최적모드에서 복원된 잔차신호의 상관성을 나타낸 그림이다. 복원된 잔차신호는 8x8 픽셀의 블록 단위로 각 절대값을 모두 합하여 도시하였다. 값이 작을수록 흰색에서 검은색으로 나타내었다. 이를 통해 복원된 잔차신호의 절대값 합이 작은 경우에는 quadtree 로 분할하여 부호화해 볼 필요성이 낮음을 확인할 수 있었다.

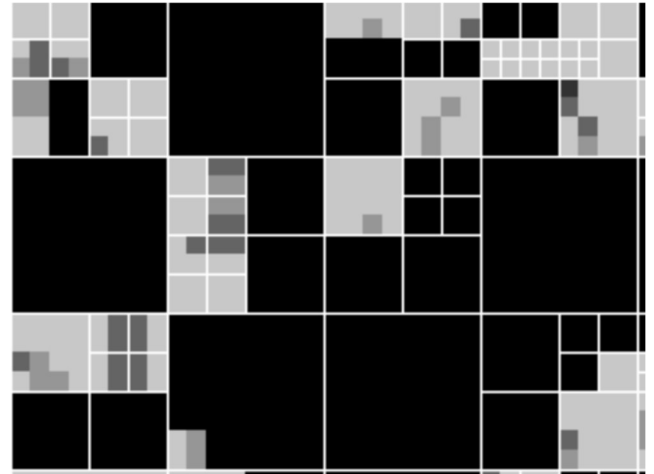


그림 5. 64x64 CTB 의 분할구조와 64x64 CB 의 최적모드에서 복원된 잔차신호의 상관성

3. 코딩 트리 블록 분할 구조 조기 결정 방법

위 실험결과를 이용한 CTB 부호화 순서도를 그림 6 에 나타내었다. 부호화 순서(i)는 표 1 을 따르며, split_flag 는 절대값의 합이 0 보다 큰 경우 1, 아닌 경우 0 으로 셋팅된다. 그림 6 을 통해서 부호화를 진행하게 되면, CB 내 split_flag 가 모두 0 인 경우에 더이상 CB 가 분할되지 않게 됨으로써 부호화기의 복잡도를 줄일 수 있다.

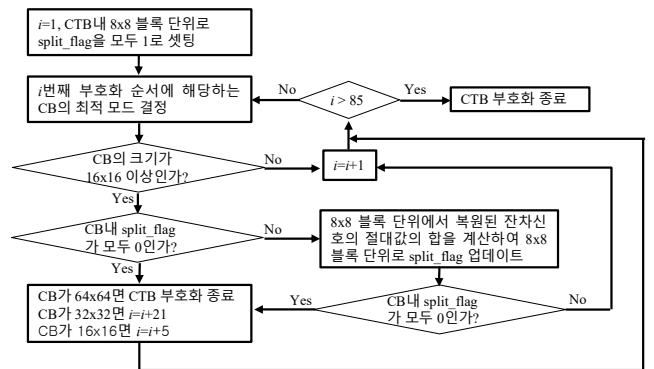


그림 6. CTB 부호화 순서도

표 1. 64x64 CTB 내 CB 부호화 순서-CTB 좌상단의 픽셀 위치를 (x,y) = (0,0)으로 설정

CB 부호화 순서(i)	x 좌표	y 좌표	CB 크기	CB 부호화 순서(i)	x 좌표	y 좌표	CB 크기
1	0	0	64x64	44	0	32	32x32
2	0	0	32x32	45	0	32	16x16
3	0	0	16x16	46	0	32	8x8
4	0	0	8x8	47	8	32	8x8
5	8	0	8x8	48	0	40	8x8
6	0	8	8x8	49	8	40	8x8
7	8	8	8x8	50	16	32	16x16
8	16	0	16x16	51	16	32	8x8
9	16	0	8x8	52	24	32	8x8
10	24	0	8x8	53	8	48	8x8
11	8	16	8x8	54	8	56	8x8

12	8	24	8x8	55	0	48	16x16
13	0	16	16x16	56	0	48	8x8
14	0	16	8x8	57	8	48	8x8
15	8	16	8x8	58	0	56	8x8
16	0	24	8x8	59	8	56	8x8
17	8	24	8x8	60	16	48	16x16
18	16	16	16x16	61	16	48	8x8
19	16	16	8x8	62	24	48	8x8
20	24	16	8x8	63	16	56	8x8
21	16	24	8x8	64	24	56	8x8
22	24	24	8x8	65	32	32	32x32
23	32	0	32x32	66	32	32	16x16
24	32	0	16x16	67	32	32	8x8
25	32	0	8x8	68	40	32	8x8
26	40	0	8x8	69	32	40	8x8
27	32	8	8x8	70	40	40	8x8
28	40	8	8x8	71	48	32	16x16
29	48	0	16x16	72	48	32	8x8
30	48	0	8x8	73	56	32	8x8
31	56	0	8x8	74	48	40	8x8
32	48	8	8x8	75	56	40	8x8
33	56	8	8x8	76	32	48	16x16
34	32	16	16x16	77	32	48	8x8
35	32	16	8x8	78	40	48	8x8
36	40	16	8x8	79	32	56	8x8
37	32	24	8x8	80	40	56	8x8
38	40	24	8x8	81	48	48	16x16
39	48	16	16x16	82	48	48	8x8
40	48	16	8x8	83	56	48	8x8
41	56	16	8x8	84	48	56	8x8
42	48	24	8x8	85	56	56	8x8
43	56	24	8x8				

4. 실험 결과

성능을 확인하기 위해 HM16.0 에 제안된 방법을 구현하여 비교 실험을 진행하였다. 표 2 와 같이 1920x1080 해상도를 가지는 5 개의 영상을 random-access 및 low-delay configuration 으로 4 가지 QPI 에 대해서 각각 100 장씩 부호화하여 ATS(average time saving) 및 BD-rate_{YUV} 를 측정하였다. ATS는 식 1로부터 계산되며, BD-rate_{YUV}는 Y, U, V 에 대한 PSNR 을 고려하여 식 2로부터 얻어진 PSNR_{YUV} 를 이용하여 얻어진 BD-rate 값이다[6].

표 2. 테스트 영상 부호화 정보

Sequence	Resolution	Frame rate	IDR period	Encoding	QPI
Kimono	1920x1080	24	24	100	22, 27, 32, 37
PakScene	1920x1080	24	24	100	
Cactus	1920x1080	50	48	100	
BQTerrace	1920x1080	60	64	100	
BasketballDrive	1920x1080	50	48	100	

$$ATS = \frac{\text{부호화 시간(HM16.0)} - \text{부호화 시간(제안 방법)}}{\text{부호화 시간(HM16.0)}} \times 100\% \quad (1)$$

$$PSNR_{YUV} = (6 \cdot PSNR_Y + PSNR_U + PSNR_V) / 8. \quad (2)$$

표 3 은 실험 결과를 보여준다. Random-access 의 경우 평균 50.98%의 부호화 시간이 줄었고, 이때 BD-rate_{YUV} 는

표 3. 실험 결과

	Random-access			Low-delay	
	QPI	ATS	BD-rate _{YUV}	ATS	BD-rate _{YUV}
Kimono	22	25.37%	2.12	16.72%	1.86
	27	37.11%		26.42%	
	32	49.45%		38.13%	
	37	59.35%		49.41%	
ParkScene	22	45.20%	2.46	35.76%	2.89
	27	57.59%		48.43%	
	32	65.86%		58.41%	
	37	70.88%		65.63%	
Cactus	22	34.25%	3.18	25.87%	2.99
	27	47.78%		38.81%	
	32	56.52%		49.39%	
	37	62.88%		57.47%	
BQTerrace	22	33.23%	1.89	27.00%	2.02
	27	57.61%		51.77%	
	32	68.72%		66.37%	
	37	74.25%		73.31%	
BasketballDrive	22	28.20%	2.44	20.41%	1.97
	27	39.22%		31.45%	
	32	49.09%		42.64%	
	37	57.06%		51.95%	
Avg.		50.98%	2.42	43.77%	2.35

2.42%로 나타났다. Low-delay 의 경우에는 43.77%의 부호화 시간 감소와 2.35%의 BD-rate_{YUV} 가 측정되었다. 모든 테스트 영상에 대해서 QPI 가 증가하면서 ATS 도 증가하는데, QPI 가 증가할수록 복원된 잔차신호의 값이 작아지기 때문이다. BQTerrace 의 경우 QPI 37 에서 부호화 시간을 74.25%까지 줄일 수 있음을 확인할 수 있었다.

5. 결론

본 논문에서는 HEVC 부호화시 코딩 트리 블록의 분할 구조를 고속으로 결정하는 방법을 제안하였다. 제안하는 방법에서는 부호화 과정에서 복원된 잔차신호가 코딩 트리 블록의 분할 구조와 상관성이 있음을 실험을 통해 확인하였고, 이를 기반으로 코딩 트리 블록의 분할 구조를 고속으로 결정하는 방법을 제안하였다. 실험을 통해 제안하는 방법이 random-access 및 low-delay configuration 에서 HM16.0 에 비해 각각 50.98%와 43.77%의 부호화 시간을 줄일 수 있음을 보여주었다. 이때 BD-rate_{YUV}는 각각 2.42%, 2.35%로 부호화 효율에는 크게 영향을 미치지 않는 것을 확인할 수 있었다.

감사의 글

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0101-16-0295, 초고품질 콘텐츠 지원 UHD 실감방송/디지털 시네마/사이니지 융합 서비스 기술 개발)

참고문헌

[1] High Efficiency Video Coding, ITU-T Rec. H.265 and ISO/IEC 23008-2, ITU-T and ISO/IEC JTC 1, Apr. 2015.

[2] S. H. Jung, H. W. Park, “A Fast Mode Decision Method in HEVC Using Adaptive Ordering of Modes”, *IEEE Trans. Circuit and Systems for Video Technology*, vol. 26, no. 10, pp. 1846–1858, Oct. 2016.

[3] L. Shen, Z. Zhang, and Z. Liu, “Effective CU size decision for HEVC intracoding,” *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.

[4] J. Xiong, H. Li, O. Wu, and F. Meng, “A fast HEVC inter CU selection method based on pyramid motion divergence,” *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[5] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, G. Sullivan, “High Efficiency Video Coding (HEVC) Encoder Description v 16 (HM16)”, *JCT-VC Document*, JCTVC-R1002, July, 2014.

[6] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves”, *ITU-T Q6/SG16, Doc. VCEG-M33*, Apr. 2001.