

정규논문 (Regular Paper)

방송공학회논문지 제19권 제3호, 2014년 5월 (JBE Vol. 19, No. 3, May 2014)

<http://dx.doi.org/10.5909/JBE.2014.19.3.405>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

초기 CU 크기 예측과 PU 모드 예측 비용을 이용한 고속 CU 결정 알고리즘

유향미^{a)}, 신수연^{a)}, 서재원^{a)†}

Fast CU Decision Algorithm using the Initial CU Size Estimation and PU modes' RD Cost

Hyang-Mi Yoo^{a)}, Soo-Yeon Shin^{a)}, Jae-Won Suh^{a)†}

요 약

HEVC는 재귀적 쿼드 트리 구조를 갖는 CU를 부호화에 적용함으로써 높은 부호화 효율을 얻었다. 그러나 이러한 재귀적 쿼드 트리 구조는 HEVC의 부호화 복잡도를 매우 증가시키는 결과를 가져왔다. 본 논문에서는 이러한 재귀적 쿼드 트리 구조 안에서 빠른 CU 결정이 가능한 알고리즘을 제안한다. 제안하는 알고리즘은 CTU 부호화가 이루어지기 전에 미리 초기 CU 크기를 예측하고, CU 부호화 과정에서 CBF와 PU 모드 예측 비용을 이용한 조건을 확인하여 고속 CU 결정이 이루어지도록 한다. 또한 인터 PU 모드 예측 과정에서 얻은 CBF값들을 이용하여 인트라 모드 예측 생략이 가능하다. 실험결과, 제안한 알고리즘의 조건에 포함된 가중치값에 따라 최대 평균 49.91%, 37.97%의 부호화 시간 감소 효과를 얻을 수 있었다.

Abstract

High Efficiency Video Coding(HEVC) obtains high compression ratio by applying recursive quad-tree structured coding unit(CU). However, this recursive quad-tree structure brings very high computational complexity to HEVC encoder. In this paper, we present fast CU decision algorithm in recursive quad-tree structure. The proposed algorithm estimates initial CU size before CTU encoding and checks the proposed condition using Coded Block Flag(CBF) and Rate-distortion cost to achieve the fast encoding time saving. And, intra mode estimation is also possible to be skipped using the CBF values acquired during the inter PU mode estimations. Experiment results shows that the proposed algorithm saved about 49.91% and 37.97% of encoding time according to the weighting condition.

Keyword : HEVC, fast encoding time, CBF, Rate-distortion cost

a) 충북대학교 전자공학과 (Dept. of Electronic Engineering, Chungbuk National University)

† Corresponding Author : 서재원(Jae-Won Suh)

E-mail: sjwon@cbnu.ac.kr

Tel: +82-43-261-3268

※ 이 논문은 2012학년도 충북대학교 학술연구지원사업에 의하여 연구되었음.

· Manuscript received April 17, 2014 Revised May 23, 2014 Accepted May 23, 2014

1. 서 론

현재 고화질의 고해상도 HD 콘텐츠와 HD 방송 서비스가 사용자에게 제공되고 있다. 곧, 더 높은 화질에 대한 사용자의 요구에 따라 초고화질의 UHD 콘텐츠와 UHD 방송 서비스가 제공될 것이다. 이러한 UHD의 서비스를 제공하

기 위해서는 비디오 압축 표준 기술인 HEVC의 기술이 필요하다. 이 표준 기술은 H.264/AVC의 두 배 압축률을 갖는 것을 목표로 국제 표준화 단체인 ITU-T와 SO/IEC MPEG의 공동 작업으로 JCT-VC팀에서 표준화가 이루어졌다^[1]. 표준화는 현재 마무리되었으며, H.264/AVC의 참조 소프트웨어 JM 18.4와 HEVC의 HM 9.0과 비교할 때 HEVC는 약 35% 정도의 압축 효율을 이루어냈다^[2].

HEVC는 H.264/AVC의 기반으로 이루어졌기 때문에 비슷한 코딩 구조를 갖는다. 그러나 H.264/AVC는 고정된 16×16 크기의 매크로 블록을 기본 단위로 갖는 반면에, HEVC는 가변적으로 크기 조절이 가능한 CTU(Coding Tree Block)를 기본 단위로 갖으며, 영상안의 슬라이스를 가장 큰 CU (Coding Unit)크기로 분할한 CTU의 번호 순서로 부호화가 진행된다. 각각의 CTU는 재귀적인 쿼드 트리 구조에 의해 더 작은 CU들로 나뉘며, 이러한 CU는 한 개의 PU (Prediction Unit) 또는 그 이상의 PU들에 의해 예측이 되고, TU(Transform Unit)의 단위로 변환(Transform)이 수행된다. 이러한 TU 역시 재귀적인 쿼드 트리의 구조를 갖는다. 따라서 다양한 크기의 CU, PU, TU가 존재한다. 재귀적인 쿼드 트리 구조는 HEVC가 높은 코딩 효율을 갖게 만들지만 또한 HEVC가 상당히 높은 복잡도를 갖도록 만드는 결과를 가져왔다. 이러한 높은 복잡도는 영상의 빠른 처리와 실시간 처리에 문제가 될 수 있다.

HEVC의 복잡도 감소를 위한 고속 CU 결정 알고리즘 기법들이 발표되었다^{[5]-[12]}. Gwen^{[5],[6]}은 HEVC의 복잡도 감소를 위해 CU 부호화의 조기 종료 알고리즘 (early termination of CU encoding)을 제안하였다. 만약 현재 PU 모드의 예측을 끝냈을 때 휘도신호와 색차신호의 CBF(Coded Block Flag)가 모두 0이라면, 현재 CU 부호화 과정에 남아 있는 PU 모드들의 예측을 모두 생략한다. Choi^{[7],[8]}도 쿼드 트리 구조를 갖는 CU 부호화 과정에서의 조기 종료 알고리즘 (coding tree pruning based early termination)을 제안하였다. 현재의 CU 부호화 과정을 수행한 후 최적의 모드가 SKIP 모드로 결정되었다면, 남아있는 하위 CU들의 부호화 과정이 생략된다. Yang^{[9],[10]} 또한 조기 스킵 모드 검출 (early SKIP mode detection)을 통한 고속 CU 결정 알고리즘을 제안하였다. CU 부호화 과정에서 Inter 2N×2N 모드

예측을 가장 먼저 실행한 후 조기 스킵 모드 조건인 움직임 벡터값의 차(DMV)와 CBF가 모두 0 이라면, 현재 CU안의 남아있는 모든 PU 모드들의 예측과 하위 CU들의 부호화 과정이 생략된다. 위의 Gwen^[5]과 Choi^[7]의 알고리즘은 HEVC의 참조 소프트웨어 HM 4.0에서부터 구현되었으며, Yang^[9]의 알고리즘은 HM 7.0에서부터 구현되었다. Kim^[11]과 Yoo^[12]는 현재 CU와 상위 CU, 현재 CU와 주변 CU들간의 통계적 특성을 이용하여 CU 부호화 시간을 감소시키는 방법을 제안하였다. 위의 알고리즘들은 무조건적으로 가장 큰 크기의 CU(CTU)로부터 가장 작은 CU (SCU:Smallest Coding Unit)들의 순서로 CU 부호화 과정을 수행하면서 빠른 CU 부호화 시간을 갖는다.

본 논문에서는 초기 CU 크기를 예측하는 방법, CBF와 각 PU 모드들의 비용을 이용한 방법으로 이루어져있는 고속 CU 결정 알고리즘을 제안한다. 본 논문의 기술 구조는 아래와 같다. 2장에서는 HEVC의 재귀적인 쿼드 트리 구조의 CU 구조를 설명하고, 3장에서는 제안하는 고속 CU 결정 알고리즘을 자세히 기술한다. 다음 4장에서는 제안하는 알고리즘의 성능 검증을 위하여 제안한 알고리즘과 기존의 다른 고속 CU 결정 알고리즘들의 실험결과를 비교하여 보인다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

II. HEVC의 모드 결정

HEVC는 영상의 시간적, 공간적 중복성을 감소시킴으로써 압축 효율을 높인다. 이를 위하여 이전의 비디오 압축 표준 기술인 H.264/AVC와 마찬가지로 하이브리드 코딩 구조를 사용하였고, 따라서 H.264/AVC와 비슷한 부호화 구조와 기술들을 갖는다. 그러나 HEVC에 향상된 부호화 기술들과 새롭게 제안된 부호화 기술들이 적용됨으로서 H.264/AVC보다 높은 부호화 효율을 얻었다. 특히 쿼드 트리 구조를 갖는 CU는 HEVC의 부호화 효율을 매우 높였고, 부호화 구조에 많은 변화를 가져왔다. 본 장에서는 이러한 쿼드 트리 구조를 갖는 CU와 이러한 재귀적인 쿼드 트리 구조 안에서 최적의 CU를 결정하는 방법을 설명한다.

1. 재귀적인 쿼드 트리 구조의 CU

HEVC에서 한 개의 영상은 CTU의 번호 순서로 부호화가 이루어진다. CTU는 사용자에게 의해 가변적으로 64×64 - 8×8까지 크기가 정해질 수 있으며 HEVC의 참조 소프트웨어에서 초기 CTU 크기는 64×64로 설정되어 있다(HM5.0). 따라서 최대 64×64 크기의 CU가 최적의 CU 크기로 결정되는 것이 가능하여 해상도가 높은 영상에서 높은 압축 효율을 얻는 것이 가능하다.

이러한 CTU는 재귀적인 쿼드 트리 구조에 의해 그림 1과 같이 더 작은 복수 개의 CU로 분할 가능하다. 한 개의 CU는 4개의 똑같은 크기 CU들로 분할하며, 4개의 CU로 분할할 때마다 그림 1과 같이 CU 깊이가 하나씩 증가된다. 이러한 분할은 재귀적인 방법으로 이루어지며, 쿼드 트리 구조를 갖게 만든다. CU의 깊이는 0에서 3까지 존재한다. 따라서 CU는 64×64에서 8×8까지의 다양한 크기를 갖게 되어 영상의 특성에 맞게 부호화가 가능하다.

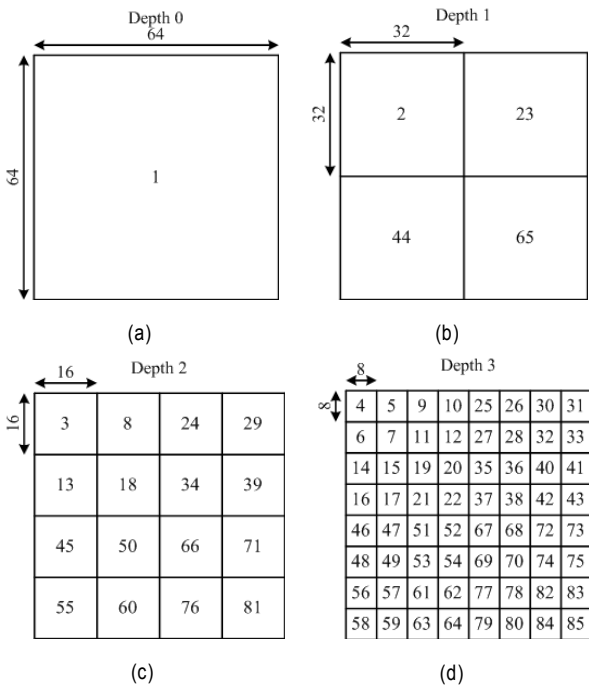


그림 1. 쿼드 트리 구조의 CU 분할
 Fig. 1. Quad-tree structured CU splitting

PU는 이러한 CU를 예측하여 CU 단위로 부호화가 이루어지게 한다. HEVC의 PU 모드는 그림 2에서 보는 것과 같이 다양한 모양의 인터 모드들과 인트라 모드로 이루어져 있다. 한 개의 CU를 부호화하기 위해 Merge/SKIP, Inter 2N×2N, Inter N×2N, Inter 2N×N, Inter 2N×nU, Inter 2N×nD, Inter nL×2N, Inter nR×2N, Intra 2N×2N, Intra N×N, Intra PCM모드가 모두 수행되어 최적의 PU 모드가 선택된다. 따라서 한 개의 CU는 인터 모드로 예측되거나 인트라 모드로 예측된다. 이렇게 예측된 최적의 PU 모드를 이용하여 복원된 블록을 얻고, 현재 블록과 복원 블록의 차이 값인 잔여 신호는 TU 단위로 변환되고 부호화된다.

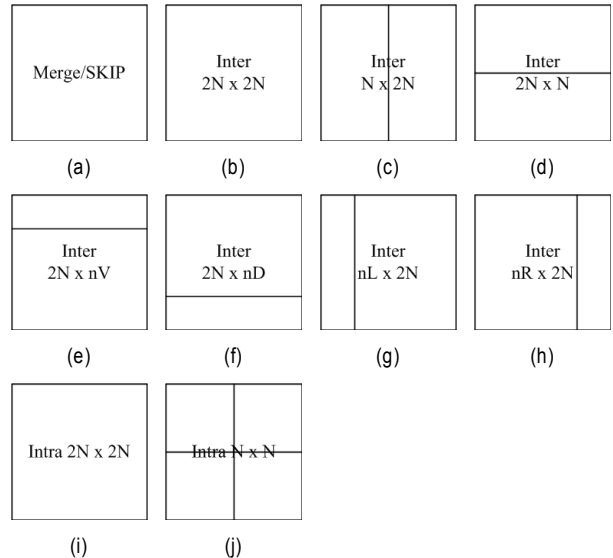


그림 2. PU 예측 모드 종류
 Fig. 2. An assortment of PU prediction mode

2. 최적의 CU 결정

최적의 CU를 결정하기 위해서는 그림 1의 계층적 구조 안에서 가장 높은 계층, 즉 CU 깊이 0에서부터 가장 밑의 깊이 3까지의 CU들의 비용을 모두 조사하여 최적의 CU가 선택된다. HEVC의 참조 소프트웨어에서의 최적의 CU 결정 과정은 다음과 같다. 먼저, 한 개의 CTU에 대한 CU 비용 조사 순서를 그림 1의 블록 번호로 나타내었다. 현재 CU

의 비용을 계산 후, 한 깊이 밑의 하위 CU의 비용을 계산하는데, 4개의 하위 CU 비용 계산이 끝날 때마다 4개의 하위 CU 비용의 합과 해당 하위 CU들의 상위 CU 1개의 비용을 비교하여, 더 작은 비용을 갖는 CU 크기가 일단은 임시적으로 최적의 CU 크기가 된다. 이러한 방법으로 한 개의 CTU에 대한 비용 계산이 모두 끝나면 임시적인 최적의 CU들이 최종적으로 최적의 CU가 된다.

예를 들어, 7번 CU까지의 비용 계산이 끝났을 때, 4~7번의 CU 비용의 합과 3번 CU의 비용을 비교하여, 만약 3번 CU 비용이 더 작은 값을 갖는다면, 임시적으로 3번 CU가 최적의 CU로 결정된다. 이러한 방법으로 마지막 85번 CU까지 비용 계산이 끝났을 때, 82~85 CU들의 비용 합과 81번 CU의 비용을 비교하여 임시적인 최적의 CU를 구한다. 만약 81번이 임시적으로 최적의 CU가 되었다면, 81번의 비용은 다시 66, 71, 76번 CU 위치의 임시적인 최적의 비용들과 합하여, 65번 CU의 비용과 비교된다. 여기서 더 작은 비용을 갖는 CU는 2, 23, 44번 위치의 임시적인 최적의 CU 비용과 합하여 마지막으로 한 번 더 1번 CU의 비용과 비교한다. 둘 중 더 적은 비용을 갖는 CU가 최종적으로 최적의 CU로 결정된다. 이처럼 한 개의 CTU에 대한 최적의 CU 결정을 위해서는 많은 계산 처리가 필요하다.

3. CU의 비용 계산

최적의 CU를 결정하기 위한 CU 비용 계산은 현재 CU에 대해서 그림 2의 모든 PU 모드 예측이 수행되고 이 중에서 가장 적은 비용을 갖는 PU 모드가 최적의 PU 모드로 선택되어 이루어진다. 또한 각각의 PU 모드 예측 수행 과정에서는 해당 PU 모드를 이용해서 얻은 현재 CU 크기의 복원 블록과 현재 CU 블록 간의 차를 이용하여 비용을 계산한다. 비용을 계산하는 방법은 아래의 식 (1), (2)와 같다.

$$SSE_{mode} = \sum_{i,j} (OrgBlock(i,j) - RecBlock(i,j))^2 \quad (1)$$

$$RD_{mode} = SSE_{mode} + \lambda_{mode} R_{mode} \quad (2)$$

SSE_{mode} 는 원영상의 블록영상인 $OrgBlock$ 과 원영상의 블

록영상이 부호화와 복호화 거쳐져서 복원된 블록 영상 $RecBlock$ 간의 차의 제곱 값이다. RD_{mode} 값은 이러한 SSE와 라그랑지안 계수인값인 λ , 그리고 현재 CU를 현재 PU 모드를 이용해서 인코딩할 때 필요한 비트수인 R 을 이용해서 계산되어진다.

III. 제안하는 알고리즘

본 논문에서 제안하는 고속 CU 결정 알고리즘은 초기 CU 크기 예측 방법, CBF와 PU 모드 예측 비용을 이용한 고속 CU 모드 결정 방법, 인트라 모드 예측 생략 방법으로 이루어져있다. 전체적인 알고리즘의 순서도는 그림 3에서 나타낸다.

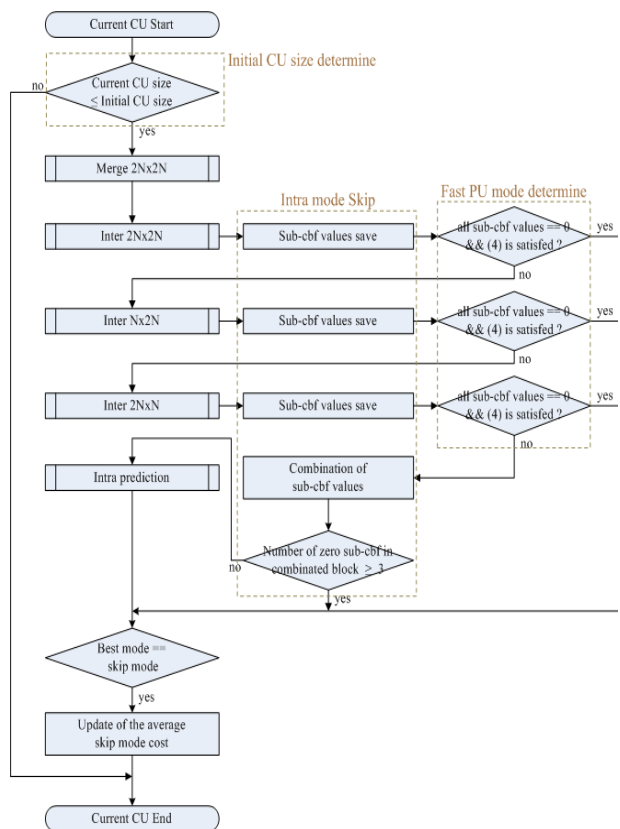


그림 3. 제안하는 알고리즘의 순서도
Fig. 3. Flowchart of the proposed algorithm

1. 초기 CU 크기 결정

HEVC에서는 최적의 CU를 찾기 위해 가장 큰 크기의 CTU 비용 조사를 시작으로 가장 작은 크기의 CU까지의

표 1. CU 크기별 CU 점유율
 Table 1. CU Depth Possession Rate

영상 크기	실험 영상	QP	각 CU크기에 따른 CU 점유율 (%)			
			64×64	32×32	16×16	8×8
			Depth 0	Depth 1	Depth 2	Depth 3
2560×1600	Traffic	22	37.33	29.26	21.36	12.04
		27	56.14	24.25	13.38	6.21
		32	67.60	19.88	9.07	3.44
		37	75.68	16.42	6.08	1.80
	People On Street	22	7.90	21.53	37.56	32.98
		27	14.21	25.84	36.92	23.01
		32	20.95	31.46	33.76	13.81
		37	29.56	35.14	27.59	7.69
1920×1080	Park Scene	22	32.89	28.06	23.03	16.00
		27	50.41	24.05	16.10	9.42
		32	62.22	20.55	11.78	5.42
		37	71.49	17.60	8.09	2.81
	BQ Terrace	22	17.24	28.42	28.72	25.61
		27	52.71	27.27	13.19	6.81
		32	72.41	17.08	7.11	3.38
		37	80.33	12.44	4.95	2.26
	Basketball Drive	22	37.71	30.73	19.22	12.32
		27	53.12	26.30	14.54	6.01
		32	62.04	23.29	11.13	3.52
		37	68.66	20.97	8.16	2.20
832×480	Basketball Drill	22	28.21	31.55	21.28	18.93
		27	38.35	29.34	19.70	12.58
		32	46.56	29.13	17.28	7.01
		37	53.35	29.71	13.21	3.72
	Party Scene	22	13.61	30.76	29.20	26.41
		27	25.98	32.66	23.38	17.96
		32	40.50	30.30	18.00	11.18
		37	54.02	26.54	13.11	6.32
	Race horses	22	7.88	28.03	36.55	27.52
		27	15.42	34.29	31.79	18.48
		32	24.88	38.83	25.38	10.89
		37	35.53	40.03	18.56	5.87
416×240	Basketball Pass	22	13.12	27.58	30.16	29.13
		27	16.82	31.28	30.89	20.99
		32	21.80	35.43	29.63	13.11
		37	28.20	39.00	25.51	7.27
	BQ Square	22	9.44	31.64	34.73	24.16
		27	24.60	40.27	22.35	12.76
		32	46.08	31.84	14.74	7.32
		37	56.73	26.09	12.45	4.72
	Race Horses	22	3.17	16.94	38.47	41.40
		27	5.42	26.14	39.42	29.00
		32	9.85	35.47	37.86	16.80
		37	18.55	43.92	29.48	8.03

모든 비용을 계산한다. 본 논문에서는 부호화가 끝난 후 각각의 CU크기에 따른 CU 선택율을 알아보기 위해 실험을 수행하였다. 이 실험에서는 여러 가지의 QP값(22, 27, 32, 34)과 여러 가지 해상도를 갖는 11가지의 실험 입력 영상들(Traffic, PeopleOnStreet, ParkScene, BQTerrace, BasketballDrive, BasketballDrill, PartyScene, RaceHorses, BasketballPass, BQSquare, RaceHorses)을 사용하였고, high efficiency - random access의 조건에서 비대칭 파티션 PU 모드를 사용하지 않고 실험하였다. 이 실험의 결과는 표1에 보인다. 실험결과, 각각의 CU 선택율은 QP와 입력 영상의 해상도, 입력 영상들의 특성에 따라 많은 영향을 받는 것을 알 수 있었다. QP값과 입력 영상의 해상도가 낮고, 입력 영상이 빠르고 복잡한 움직임을 보이는 경우 CTU가 최적의 CU 크기로 결정되는 확률이 낮았다. 따라서 무조건적으로 가장 큰 CTU의 비용 조사를 시작으로 하는 CU 부호화는 부호화 시간을 낭비할 수 있다. 그러므로 CU 부호화가 시작되기 전에 미리 입력 영상의 공간적, 시간적 특성을 적응적으로 이용하여 초기의 CU 크기를 예측한다면 CU 부호화 시간을 감소시킬 수 있다.

공간적 영역에서 인접한 영역 간에는 높은 공간적 상관성이 존재할 가능성이 높으며, 동영상에서 시간적으로 가까운 프레임과는 높은 시간적 상관성을 갖을 가능성이 높다. 따라서 초기 CU 크기를 예측하기 위해서 그림 4와 같이 현재 CTU의 주변 CU들과 이전 프레임에서의 현재 CTU 위치 CU들의 크기를 조사한다. 이러한 초기 CU 크기의 값이 값은 아래의 식 (3)과 같이 구할 수 있다.

$$Initial\ CUdepth = Med\{A, B, C, D, E\} - 1\ depth \quad (3)$$

식 (3)의 A, B, C, D는 그림 4와 같이 현재 CTU의 왼쪽, 왼쪽 위, 위, 오른쪽 위로 접해있는 CU들 중에서 A위치에서 가장 큰 CU의 깊이 값, B위치에서 가장 큰 CU의 깊이 값, 마찬가지로 C, D 위치에서의 가장 큰 CU의 깊이 값을 각각 나타낸다. 식(3)의 E는 바로 이전에 부호화된 프레임에서 현재 위치의 CTU 안에서 가장 큰 CU의 깊이 값을 말한다. 각각의 위치에서 가장 큰 CU의 깊이 값들 중에서 중간 값을 선택하고, 이 깊이 값에서 한 깊이 값을 빼면 초

기 CU 깊이 값이 결정된다. 식 (3)에서 A~E값들은 각각의 위치에서 가장 큰 CU의 깊이 값들이므로 이 값들 중에서 최대 값을 선택하면 대부분 가장 큰 64×64 CU가 시작 CU로 결정되기 때문에 빠른 부호화 시간을 얻기 위해 중간 값을 선택한다.

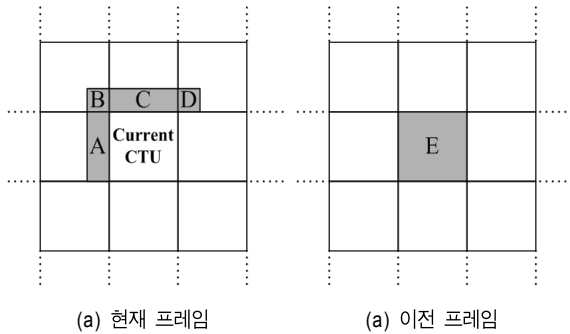


그림 4. 초기 CU 크기 예측
Fig. 4. Initial CU size estimation

2. CBF와 PU모드 비용을 이용한 고속 CU 결정

최적의 CU를 결정하기 위해서는 전에 언급한 것과 같이 그림 2의 모든 PU 모드 종류에 대하여 비용을 계산한다. 그 중 가장 작은 비용을 갖는 최적의 PU 모드를 찾기 위해 모든 PU 모드의 비용은 순서대로 다음 PU 모드의 비용과 비교된다. 이러한 구조 안에서 계산적 복잡도의 감소를 위해 본 장에서는 CBF와 PU 모드의 비용을 이용한 고속 CU 결정 방법을 제안한다. 각 PU 모드의 비용과 CBF 값, SKIP 모드의 비용을 조사하였다. PU 모드 예측 후에 CBF가 0이라면 부호화될 잔여 신호가 존재하지 않기 때문에 작은 비용을 갖으며, 따라서 SKIP 모드는 작은 비용을 발생시킨다. 이러한 조건을 이용하여 고속 CU 결정 방법을 제안한다. 제안하는 방법은 그림 2의 2N×nU, Inter 2N×nD, Inter nL×2N, inter nR×2N와 같은 비대칭 파티션 모드는 사용하지 않는다.

제안하는 방법은 인터 모드 예측 수행이 끝날 때마다 두 가지의 조건을 확인하고 두 가지의 조건이 모두 만족하면, 현재 CU 내에 남아있는 나머지 PU 모드 예측들을 모두 생략한다. 또한 현재 CU 위치의 하위 CU들의 부호화 과정도

생략한다. 첫 번째 조건은 현재 PU 예측 모드의 CBF값이 0인지 확인하는 것이며, 두 번째 조건은 현재 PU 모드의 비용이 식(5)를 만족하는지 확인하는 것이다. 현재 CU의 부호화를 위해 Merge 예측 모드가 수행된 후 인터 2N×2N 모드 예측이 끝나게 되면, 먼저 휘도 신호와 색차 신호의 CBF값이 모두 0인지 확인하고, 현재 PU 예측 비용을 식(5)의 WAvgRDCost_{SKIP}과 비교하여 현재 PU 비용이 더 작은 값을 갖는지 확인한다. 이러한 조건들이 만족 한다면, 현재 CU에서 각 PU 모드들을 차례로 예측 하는 과정을 끝내고, 현재 CU의 하위 CU들의 부호화 과정도 수행하지 않는다. 그러나 위의 두 조건을 모두 만족하지 않는다면 인터 N×2N 모드 예측을 수행하고 위의 두 가지 조건을 다시 확인한다. 그러나 이번에도 만족하지 않는다면 인터 2N×N 모드 예측 후에 현재 CU 부호화 과정에서 마지막으로 한번 더 마찬가지로 조건을 확인한다. WAvgRDCost_{SKIP} 값은 아래의 식 (4)와 같이 계산한다.

$$WAvgRDCost_{SKIP} = \alpha \cdot \sum_n^{n-1} RDCost_{SKIP} / 5 \quad (4)$$

$$RDCost_{mode} < WAvgRDCost_{SKIP} \quad (5)$$

WAvgRDCost_{SKIP}는 CU 크기에 따라 각각의 값이 따로 계산되어 존재하며, 이 값은 스킵 모드가 최적의 모드로 선택될 때마다 가장 최근의 스킵 모드 비용 4개와 현재 스킵 모드의 비용을 평균낸 값에 α를 곱하여 갱신된다. 제안하는 알고리즘의 조건에 α를 적용하여 CU의 부호화 속도 조절이 가능하도록 하였다. 본 장에서 제안하는 방법은 정확하고 빠르게 최적의 PU 모드를 찾기 위해 CBF와 PU 예측 비용을 사용하여 균일한 특성으로 이루어진 영역의 PU 모드를 빨리 걸러내려 하였다. 따라서 Gweon[5]의 CBF값을 확인하는 조건이 사용되었다. Gweon[5]은 CBF 조건이 만족되어도 다음 하위 CU들의 부호화 과정을 수행하면서 다시 조건을 확인하는 반면, 제안하는 방법에서는 조건이 만족하면 남아있는 다음 PU 모드들의 예측과 현재 CU의 하위 CU들의 부호화 과정을 모두 생략한다.

제안하는 알고리즘을 그림 1을 이용해서 예를 들면, CU 깊이 1의 2번 CU의 인터 2N×2N 예측 후에 위의 제안하는

방법의 조건을 모두 만족한다면 현재 CU에 남아있는 다음 인터 2N×N 모드, 인터 N×2N 모드, 인트라 모드 예측은 모두 수행되지 않는다. 또한 남아있는 CU 깊이 2의 3, 8, 13, 18번의 하위 CU들의 부호화와 CU 깊이 3의 4~7, 9~12, 14~17, 19~22번의 하위 CU들의 부호화는 이루어지지 않는다.

3. 인트라 모드 생략

만약 위의 CBF와 PU 모드 비용을 이용한 방법에 의해 CU 결정이 초기에 종료되지 않았다면, 인터 PU 모드 예측 과정 중에 획득한 CBF값들을 이용한 인트라 모드 생략 방법을 적용한다. 인터 PU 모드를 수행할 때 마다 4개의 N×N 크기의 CBF를 각각 저장해둔다. 여기에서는 서브 CBF라고 표현한다. 인트라 모드의 수행 여부를 결정하기 위해 그림 5에서 보이듯이 각 서브 CBF의 위치에 따라 AND 연산을 실행하여 서브 CBF 맵을 생성한다. CBF값이 0이 아닐 땐 1로 고려하여 AND연산 수행한다.

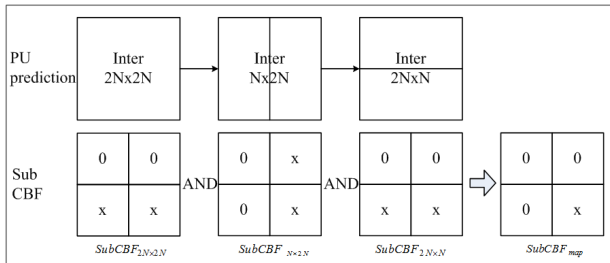


그림 5. Sub CBF 맵
 Fig. 5. Sub CBF Map

이러한 과정은 식 (6)과 같이 표현될 수 있다.

$$SubCBF_{map_i} = SubCBF_{2N \times 2N_i} \& SubCBF_{N \times 2N_i} \& SubCBF_{2N \times N_i} \quad (6)$$

$$number\ of\ zeros\ in\ SubCBF_{map_i} \geq 3 \quad (7)$$

SubCBF_{2N×2N_i}는 인터 2N×2N 모드 예측 후에 얻은 서브 CBF값들을 나타낸다. SubCBF_{N×2N_i}와 SubCBF_{2N×N_i}

는 인터 N×2N 모드 예측과 인터 2N×N 모드 예측 후에 얻은 서브 CBF값들을 각각 나타낸다. 식 (6)에서 얻은 SubCBF_{map_i} (서브 CBF 맵) 안의 0의 갯수가 식 (7)과 같이 3개 이상 존재한다면 현재 CU에 대한 인트라 모드 예측을 생략한다.

예를 들어, 현재 CU에 대해 인터 2N×2N 모드 예측을 수행할 때, 변환(Transform) 과정에서 현재 깊이 크기의 TU 변환 후 CBF값이 0이라면, 그림 5의 SubCBF_{2N×2N_i}는 각 4개의 위치에 모두 0의 값을 갖게 된다. 그러나 CBF값이 0이 아니라면, “현재 깊이 + 1” 크기의 TU 변환이 끝난 직후 CBF값을 확인하여 각각의 4개의 위치에 따른 N×N 크기의 서브 CBF값들을 얻는다. 인터 N×2N 모드와 인터 2N×N 모드는 마찬가지로 “현재 깊이 + 1” 크기의 TU 변환이 끝난 직후 위치에 따라 CBF를 확인하여 서브 CBF값들을 얻는다.

IV. 실험결과

제한한 고속 CU 모드 결정 알고리즘을 HM5.0 참조 소프트웨어에서 구현하여 성능을 평가하였다. 실험을 위해 5초의 ClassA 영상과 10초의 ClassB, C, D, E 영상이 사용되었으며 QP값은 22, 27, 32, 37을 사용하였다. 또한 random access와 low delay P의 조건에서 실험하였으며 비대칭 파티션 모드를 사용하지 않는 것을 제외한 모든 조건은 초기 조건과 동일하게 실험하였다. 제안한 알고리즘의 실험결과는 Gweon^[5]과 Choi^[7]의 실험 결과와 비교하여 보여주며, 아래의 식 (8), (9), (10)과 같은 방법을 이용하여 PSNR, 비트율, 부호화 시간을 비교한다.

$$\Delta PSNR = (PSNR_{proposed} - PSNR_{ref}) \quad (8)$$

$$\Delta Bits = (Bits_{proposed} - Bits_{ref}) / Bits_{ref} \times 100(\%) \quad (9)$$

$$\Delta Time = (EncTime_{proposed} - EncTime_{ref}) / EncTime_{ref} \times 100(\%) \quad (10)$$

표 2와 표 3에 각 알고리즘에 의해 도출된 실험결과를

표 2. Random Access의 조건에서 실험한 결과
Table 2. Experimental Results Under Random access

Size	Video	Proposed Algorithm ($\alpha: 1.0$)			Proposed Algorithm ($\alpha: 1.5$)			Gweon [5]			Choi [7]		
		ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)
2560x1600	Traffic	-0.17	-0.03	-50.37	-0.74	-0.10	-63.30	-0.55	-0.03	-45.93	-1.20	-0.06	-53.05
	PeopleOnStreet	0.58	-0.01	-25.88	0.41	-0.07	-35.03	-0.29	-0.07	-29.91	-0.64	-0.06	-24.13
1920x1080	BQTerrace	-0.50	-0.01	-47.16	-1.39	-0.05	-62.59	-0.69	-0.02	-43.80	-1.09	-0.02	-50.77
	Cactus	0.30	-0.01	-42.99	0.04	-0.04	-55.40	-0.38	-0.02	-40.23	-0.73	-0.03	-45.19
	Kimono	0.14	-0.01	-41.67	-0.20	-0.04	-56.94	-0.23	-0.01	-40.56	-0.56	-0.03	-44.89
	ParkScene	-0.09	-0.02	-47.34	-0.65	-0.07	-61.43	-0.60	-0.03	-45.05	-0.95	-0.04	-50.82
832x480	BasketballDrill	0.39	-0.00	-34.37	0.22	-0.03	-45.48	-0.42	-0.03	-34.45	-0.68	-0.03	-35.24
	BQMall	0.10	-0.02	-41.48	-0.25	-0.08	-53.93	-0.51	-0.06	-40.70	-0.82	-0.06	-43.39
	PartyScene	-0.19	-0.03	-37.03	-0.69	-0.10	-48.50	-0.60	-0.05	-35.69	-0.83	-0.05	-36.54
	Racehorses	0.33	-0.02	-27.26	0.09	-0.05	-36.18	-0.54	-0.07	-31.01	-0.50	-0.03	-24.42
416x240	BasketballPass	0.36	-0.01	-30.16	0.22	-0.05	-37.57	-0.64	-0.08	-32.63	-0.55	-0.04	-28.86
	BlowingBubbles	-0.24	-0.03	-37.74	-0.81	-0.11	-50.41	-0.68	-0.06	-36.78	-0.89	-0.05	-37.26
	BQSquare	-0.37	-0.04	-42.97	-1.18	-0.11	-57.50	-0.88	-0.05	-42.22	-0.78	-0.04	-44.40
	RaceHorses	0.26	-0.02	-25.19	-0.06	-0.08	-34.53	-0.68	-0.10	-30.34	-0.53	-0.04	-20.64
Average		0.06	-0.01	-37.97	-0.35	-0.07	-49.91	-0.54	-0.04	-37.80	-0.76	-0.04	-38.54

표 3. Low Delay의 조건에서 실험한 결과
Table 3. Experimental Results Under Low Delay

Size	Video	Proposed Algorithm ($\alpha: 1.0$)			Proposed Algorithm ($\alpha: 1.5$)			Gweon [5]			Choi [7]		
		ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)	ΔB (%)	ΔP (dB)	ΔT (%)
1920x1080	BQTerrace	-0.86	-0.02	-38.51	-2.81	-0.08	-50.42	-1.11	-0.04	-34.30	-2.21	-0.03	-38.21
	Cactus	0.57	-0.01	-33.23	0.52	-0.04	-42.99	-0.28	-0.02	-30.17	-0.68	-0.03	-32.80
	Kimono	0.32	-0.01	-30.96	0.25	-0.04	-43.85	-0.09	-0.01	-31.07	-0.32	-0.02	-31.30
	ParkScene	0.01	-0.02	-36.21	-0.40	-0.07	-48.24	-0.43	-0.03	-35.18	-0.97	-0.04	-37.58
832x480	BasketballDrill	0.52	-0.01	-26.65	0.36	-0.04	-36.22	-0.34	-0.02	-26.25	-0.58	-0.02	-26.50
	BQMall	0.27	-0.02	-32.07	0.09	-0.06	-42.25	-0.37	-0.05	-30.72	-0.56	-0.03	-31.54
	PartyScene	-0.47	-0.04	-26.81	-1.27	-0.12	-35.94	-0.74	-0.07	-22.53	-1.01	-0.04	-20.01
	Racehorses	0.50	-0.01	-22.94	0.41	-0.03	-29.88	-0.28	-0.04	-22.17	-0.33	-0.01	-18.24
416x240	BasketballPass	0.32	-0.01	-26.42	0.42	-0.03	-32.01	-0.30	-0.06	-24.81	-0.23	-0.02	-22.38
	BlowingBubbles	-0.29	-0.03	-28.63	-1.22	-0.10	-38.02	-0.66	-0.07	-27.02	-1.11	-0.04	-24.84
	BQSquare	-1.22	-0.05	-32.94	-3.68	-0.17	-42.61	-1.53	-0.10	-30.24	-2.23	-0.06	-27.45
	RaceHorses	0.47	-0.01	-21.76	0.45	-0.04	-27.74	-0.12	-0.06	-21.36	-0.11	-0.02	-15.54
1280x720	Vidyo1	0.03	-0.01	-48.38	-0.30	-0.05	-60.31	-0.42	-0.04	-40.67	-1.40	-0.03	-52.58
	Vidyo3	-0.15	-0.02	-48.60	-0.43	-0.08	-60.56	-0.44	-0.06	-40.07	-1.25	-0.05	-52.30
	Vidyo4	0.12	-0.02	-50.12	-0.32	-0.06	-61.78	-0.58	-0.04	-41.24	-1.40	-0.04	-54.07
Average		0.00	-0.01	-33.61	-0.52	-0.06	-43.52	-0.51	-0.04	-30.52	-0.95	-0.03	-32.35

정리하였다. 각 표에서 ΔB , ΔP , ΔT 의 값은 각 QP값(22, 27, 32, 37) 별로 모두 실험한 결과를 영상별로 평균을 계산하여 나타내었으며 각 영상들의 ΔB , ΔP , ΔT 값들에 평균을 취해서 각 표에 나타내었다. 표 2는 random access의 조건에서 얻은 실험결과를 나타낸다. 제안한 알고리즘과 다른 알고리즘들과의 부호화 시간을 비교하면, α 값이 1.0일 때 평균 부호화 감소 시간은 37.97% 이므로써 다른 알고리즘들과 비슷한 부호화 시간을 갖지만, PSNR값은 -0.01dB 감소로 가장 작은 변화를 갖았다. 비트율은 0.06% 증가하였다. α 값을 1.0에서 1.5로 증가시켰을 때 화질 열화는 가장 크게 발생되었지만 상당한 부호화 시간 감소를 얻을 수 있었다. 제안한 알고리즘은 전체적으로 큰 크기의 입력 영상에서 높은 부호화 시간 감소를 얻을 수 있었다.

표 3은 low delay P의 조건에서 실험한 결과를 정리하여 보인다. α 값이 1.5일 때 43.52%, α 값이 1.0일 때 33.61%의 부호화 시간 감소를 얻음으로써 다른 알고리즘들 보다 높은 부호화 시간 감소 결과를 얻었다. 평균 PSNR의 변화는 α 값이 1.0과 1.5일 때 각각 -0.1dB, -0.06dB이다. -0.01dB는 전체 알고리즘들의 PSNR 변화값 중 가장 적은 값이다. 또한 비트율은 α 값이 1.0과 1.5일 때 각각 0%와 -0.52%의 변화를 가졌다. 제안하는 알고리즘은 특히 Vidyo4(1280×720)의 영상에서 61.78%와 50.12%로 높은 부호화 시간 감소를 보였다. random access와 low delay P의 실험결과를 보면 전체적으로 큰 해상도를 갖는 영상에서 많은 부호화 시간 감소를 이루어냈다. 표 2와 3에서 보듯이 제안한 알고리즘은 전체적으로 비교적 높은 부호화 시간 감소를 보였으며 특히 PSNR 변화에서 -0.01dB로 가장 적은 변화를 보였다.

V. 결론

본 논문에서는 고속의 CU 결정 알고리즘을 제안하였다. 예측된 초기 CU 크기로 CU 부호화를 시작하여 불필요한 큰 크기의 CU 부호화를 생략할 수 있었으며 재귀적인 쿼드 트리 구조에서 CBF와 PU 모드 비용을 이용한 조건을 사용함으로써 빠른 CU 모드 결정을 이룰 수 있었다. 마지막으로

인터 PU 모드 예측 과정에서 얻은 서브 CBF값들을 이용하여 인트라 모드 예측 생략이 가능하였다. 실험결과 제안한 알고리즘은 적은 PSNR값 감소와 약간의 비트율 증가로 최대 49.91%의 상당한 부호화 시간 감소를 얻을 수 있었다.

참고 문헌 (References)

- [1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, WD9: Working Draft 9 of High Efficient Video Coding (HEVC), Document JCTVC-K1003, Shanghai, CN, October 2012.
- [2] B. Li, G. J. Sullivan and J. Xu, Comparison of Compression Performance of HEVC Draft 9 with AVC High Profile and Performance of HM9.0 with Temporal Scalability Characteristics, Document JCTVC-L0322, Geneva, CH, January 2013.
- [3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, December 2012.
- [4] K. McCann, B. Bross, I. K. Kim, S. I. Sekiguchi, W. J. Han, HM5: High Efficiency Video Coding (HEVC) Test Model 5 Encoder Description, Document JCTVC-G1102, Geneva, CH, November 2011.
- [5] R. H. Gweon, Y. L. Lee, and J. Lim, Early Termination of CU Encoding to Reduce HEVC Complexity, Document JCTVC-F045, Torino, IT, July 2011.
- [6] R. H. Gweon, Y. L. Lee, Early Termination of CU Encoding to Reduce HEVC Complexity, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E95-A, no. 7, pp. 1215-1218, July 2012.
- [7] K. Choi, S. H. Park, and E. S. Jang, Coding tree pruning based CU early termination, Document JCTVC-F092, Torino, IT, July 2011.
- [8] K. Choi and E. S. Jang, Fast coding unit decision method based on coding tree pruning for high efficiency video coding, *SPIE Optical Engineering*, vol. 51, no. 3, pp. 030502-1-030502-3, March 2012.
- [9] J. Yang, J. Kim, K. Won, H. Lee, and B. Joen, Early SKIP Detection for HEVC, Document JCTVC-G543, Geneva, CH, November 2011.
- [10] J. Kim, J. Yang, K. Won, and B. Joen, Early Determination of Mode Decision for HEVC, *Picture Coding Symposium, Kraków, Poland*, pp. 449-452, May 2012.
- [11] J. Kim, S. Jeong, S. Cho, and J. S. Choi, Adaptive Coding Unit Early Termination Algorithm for HEVC, *IEEE International Conference on Consumer Electronics, Las Vegas, USA*, pp.261-262, January 2012.
- [12] H. M Yoo and J. W. Suh, "Fast Coding Unit Decision Algorithm Based on Inter and Intra Prediction Unit Termination for HEVC," *IEEE International Conference on Consumer Electronics, Las Vegas, USA*, pp.302-303, January 2013.
- [13] Woo-Jin Han, "Conditional Probability Based Early Termination of Recursive Coding Unit Structures in HEVC", *JBE, Vol.17, No. 2*, pp. 354-362, 2012, March.

저 자 소 개



유 향 미

- 2005년 2월 : 청주대학교 통신공학과 학사
- 2009년 2월 : 충북대학교 전자공학과 석사
- 2009년 3월 ~ 현재 : 충북대학교 전자공학과 박사과정
- 주관심분야 : 동영상압축, 워터마킹



신 수 연

- 2011년 2월 : 충북대학교 전자공학과 학사
- 2013년 2월 : 충북대학교 전자공학과 석사
- 2013년 3월 ~ 현재 : 충북대학교 전자공학과 박사과정
- 주관심분야 : 멀티미디어, 워터마킹



서 재 원

- 1995년 2월 : 충북대학교 전자공학과 학사
- 1997년 2월 : 광주과학기술원 정보통신공학과 석사
- 2003년 2월 : 광주과학기술원 정보통신공학과 박사
- 2003년 4월 ~ 2004년 8월 : LG전자기술원 MM연 선임 연구원
- 2004년 9월 ~ 현재 : 충북대학교 전자정보대학 부교수, 컴퓨터정보통신연구소 비전임 연구원
- 주관심분야 : 디지털 영상신호처리, 동영상 표준화, H.264/AVC, SVC, HEVC