

논문 2012-49SP-3-5

# CU 깊이 우선 결정 기반의 HEVC 고속 부호화 방법

(Fast HEVC Encoding based on CU-Depth First Decision)

유 성 은\*, 안 용 조\*, 심 동 규\*\*

(SungEun Yoo, Yong-Jo Ahn, and DongGyu Sim)

## 요 약

본 논문은 HEVC 인코딩 속도 향상을 위한 고속 CU (Coding Unit) 결정 방법을 제안한다. 본 논문에서는 계산 복잡도 감소와 속도향상을 위하여 CU, PU (Prediction Unit), 그리고 TU (Transform Unit) 의 결정을 두 단계로 나누어 실시한다. 첫 번째 단계에서는 LCU (Largest Coding Unit)내 각 CU의 깊이를 결정하며, 이때  $2N \times 2N$  PU의 선택 비율이 높다는 통계적 특성을 고려하여  $2N \times 2N$  PU만을 사용한다. 두 번째 단계에서는 첫 번째 단계에서 결정된 CU의 깊이 정보를 이용하여, 해당 깊이에서 정확한 PU와 TU를 결정한다. 또한, 두 번째 과정에서는 보다 효율적인 복잡도 감소 효과를 얻기 위하여 제안하는 스킵을 이용한 CU 조기 종료 알고리즘을 이용한다. 제안하는 방법은 모든 깊이에서, 모든 종류의 PU와 TU의 결정을 통한 부호화 과정을 거치지 않기 때문에 계산 복잡도 감소 효과를 얻을 수 있으며, 기존의 HEVC 레퍼런스 소프트웨어인 HM3.3 대비 약 2% 정도의 비트율이 증가하면서, 약 50%의 복잡도 감소 효과를 얻을 수 있었다.

## Abstract

In this paper we propose the fast CU (Coding Unit) mode decision method. To reduce computational complexity and save encoding time of HEVC, we divided CU, PU (Prediction Unit) and TU (Transform Unit) decision process into two stages. In the first stage, because  $2N \times 2N$  PU mode is mostly selected among  $2N \times 2N$ ,  $N \times 2N$ ,  $2N \times N$ ,  $N \times N$  PU modes, proposed algorithm uses only  $2N \times 2N$  PU mode deciding depth of each CU in the LCU (Largest CU). And then, proposed method decides exact PU and TU modes at the depth level which is decided in the first stage. In addition, early skip decision rule is applied to the proposed method to obtain more efficient computational complexity reduction. The proposed method reduces computational complexity of the HEVC encoder by simplifying a CU depth decision method. We could obtain about 50% computational complexity reduction in comparison with HM 3.3 HEVC reference software while bitrate compressed by the proposed algorithm increases only 2%.

**Keywords :** HEVC, Computational complexity, Fast CU decision, Early skip, Early termination

## I. 서 론

디지털 방송기술의 비약적인 발전과 함께 최근 고해

상도, 고화질에 대한 사용자의 요구가 꾸준히 증가하고 있다. 기존의 2D를 넘어선 3D 비디오 콘텐츠의 보급과 4K×2K급 영상을 넘어서 8K×4K급 고해상도 영상에 대한 필요성도 대두 되고 있다. 뿐만 아니라 모바일 기기에서도 1080p full HD 영상 재생이 가능한 기기의 보급이 이루어지고 있다. 시간축 화면 재생율에서도 기존의 30 fps를 넘어 50~60 fps의 재생율이 요구되고 있다. 이러한 고해상도, 고화질의 비디오 콘텐츠들은 기존의 비디오 콘텐츠에 비하여 수 배에서 수십 배에 달하는 데이터량을 가지게 된다. 이러한 비디오 콘텐츠의 전송과 보관, 그리고 관리 등을 위해서는 기존의 H.264/AVC의 압축 성능을 넘어선 뛰어난 압축 효율을

\* 학생회원, \*\* 정회원-교신저자,  
광운대학교 컴퓨터공학과  
(Dept. of Computer Engineering Kwangwoon University)

※ 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)[10039199, 인지 품질 기반 스케일러블 3D 비디오 코덱 핵심 기술 연구]과 2011년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초 연구사업(NRF-2011-0027104)의 일환으로 수행하였음.

접수일자: 2011년11월28일, 수정완료일: 2012년3월20일

가지는 새로운 비디오 코덱이 필요하다. 이에 발맞추어 MPEG과 VCEG에서는 2010년 1월 JCT-VC (Joint Collaborative Team on Video Coding)를 만들어 차세대 코덱에 대한 표준화를 시작하였다. 차세대 코덱에 대한 요구사항<sup>[1]</sup>이 2010년 1월 교토회의에서 작성되어 공표되었다. 요구사항 중, 압축 성능과 복잡도에 대한 내용을 정리하자면 다음과 같다. 압축 성능에 있어 기존의 H.264/AVC의 High Profile 대비 뛰어난 압축 성능을 보여야 한다는 요구사항이 있으며, 복잡도 측면에서는 복잡도의 조정이 가능해야 한다는 점, 병렬처리가 가능해야 한다는 점이 있다. 복잡도와 압축 성능 간 상호 조절을 지원해야 한다는 것은 H.264/AVC<sup>[2]</sup>와 비교하여 더 낮은 복잡도에서 더 나은 압축 성능을 보이거나, 더 높은 복잡도에서 높은 복잡도를 보상할 수 있을 정도의 압축 성능을 보여야 한다는 의미이다.

이러한 요구사항을 가지고 JCT-VC에서는 2010년 4월 첫 회의를 시작으로 본격적인 표준화 회의를 시작하였으며, 2010년 10월 중국 광저우에서 개최된 3차 회의에서 최초의 레퍼런스 소프트웨어인 HM1.0<sup>[3]</sup>이 결정되었다. HEVC의 레퍼런스 소프트웨어인 HM과 기존의 표준 코덱인 H.264/AVC의 레퍼런스 소프트웨어인 JM의 성능을 비교하면, HM은 압축 효율면에서 약 40%의 높은 압축효율을 보이지만, 3-4배 이상의 높은 복잡도를 가진다. 압축 성능은 높지만 지나치게 높은 복잡도는 HEVC의 고효율/저복잡도의 영상 압축을 목표로 하는 HEVC의 주요 단점이라 할 수 있다. 현재 진행 중인 JCT-VC 표준화 회의에서도 HEVC의 복잡도 문제에 대한 관심이 높은 상황이며, 참여 중인 기관에서도 낮은 복잡도를 장점으로 내세우는 기고들이 주를 이루고 있는 상황이다.

HEVC 부호화기의 복잡도를 분석해 보면 CU 결정 과정이 차지하는 복잡도가 전체의 70~80%에 이른다. HEVC의 블록 구조는 하나의 LCU안에 여러 크기의 CU가 공존할 수 있고 CU는 다시 여러 크기의 PU로 나뉘어 예측을 수행하는 구조를 가지고 있다. 그렇기 때문에 하나의 LCU를 부호화 하기 위한 CU와 PU의 조합이 많고 높은 복잡도를 요구한다.

본 논문에서는 HEVC의 높은 복잡도 문제를 해결하고자 LCU안의 CU 결정 과정을 간소화 하는 고속 CU 결정 방법을 제안한다. 제안하는 방법은 크게 두 가지로, 스킵을 이용한 CU 조기 종료 알고리즘과 LCU를 부호화 하는 과정을 두 단계로 나누어 복잡도를 줄이는

방법이다. 스킵을 이용한 CU 조기 종료 알고리즘은 머지 스킵에 대한 RDcost가 일정 문턱치를 넘지 않는다면, 이후 CU 결정 단계를 생략하는 방법이다. 두 번째로 언급한 방법은 CU 결정 과정을 두 단계로 나누어 첫 번째 단계에서는 CU의 깊이를 결정하고, 두 번째 단계에서는 결정된 깊이에서 PU와 TU를 결정하는 방법이다. 제안하는 깊이 우선 결정 방법은 먼저 CU의 깊이를 결정하고 결정된 CU 깊이에서만 세밀한 예측과 변환 등을 수행하기 때문에 CU 깊이 예측이 잘 이루어진다면 심각한 화질의 열화 없이 낮은 복잡도로 부호화가 가능하다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 방법에 대하여 살펴보고, III장에서는 제안하는 방법에 대해 소개한다. IV장에서는 제안하는 방법을 검증하기 위한 실험 방법과 실험 결과에 대하여 살펴본다. 마지막으로 V장에서 결론을 맺는다.

## II. HEVC의 계층적 블록 구조와 블록 구조 결정 방법

HEVC의 레퍼런스 소프트웨어인 HM에서는 계층적 부호화 구조 및 변환 블록 구조를 위해 CU (Coding Unit), PU (Prediction Unit), TU (Transform Unit)을 정의하여 사용하며, CU의 크기와 TU의 크기는 그 최

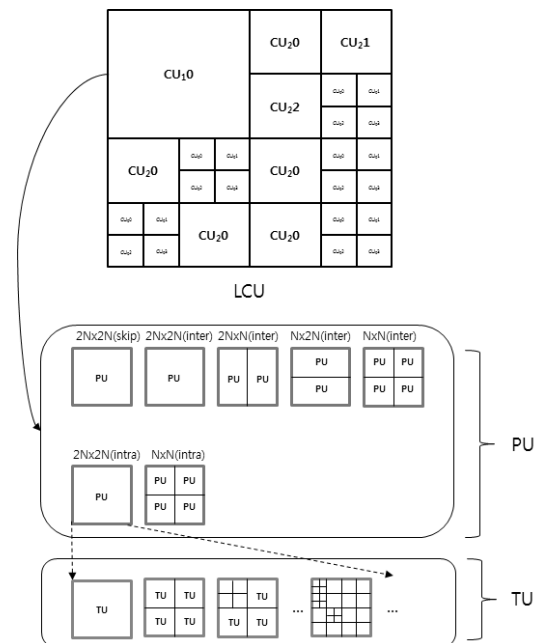


그림 1. HEVC의 계층적 블록 구조도  
Fig. 1. Hierarchical block structure of HEVC.

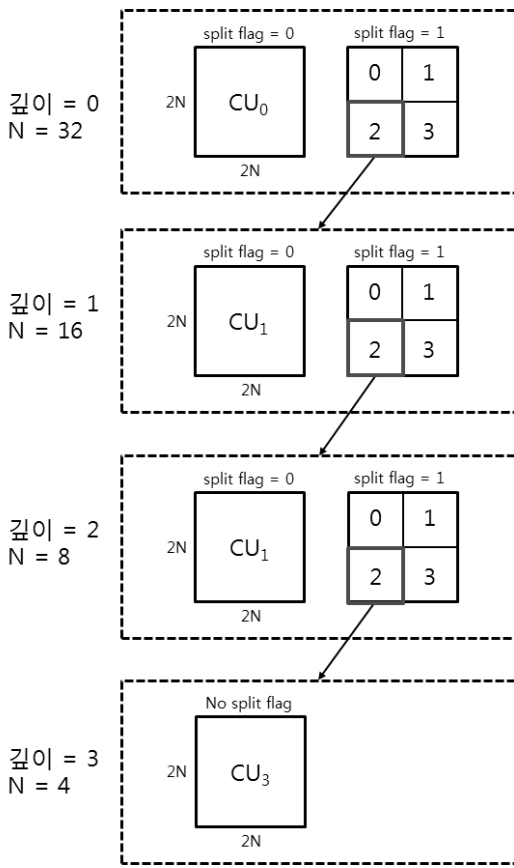


그림 2. 재귀적 CU 구조  
Fig. 2. Recursive CU structure.

맷값과 최솟값이 SPS (Sequence Parameter Set)에 포함되어 전송된다<sup>[4]</sup>. 정의하는 가장 큰 CU를 LCU, 가장 작은 크기의 CU를 SCU (Smallest Coding Unit)로 부르는데, HM 1.0에서는 LCU를 128x128크기의 블록으로, SCU를 8x8크기의 블록으로 정의 하였으며, 이 후 HM 2.0부터는 LCU의 크기를 64x64, SCU의 크기를 8x8크기의 블록으로 정의하였다.

그림 1은 HEVC의 계층적 블록 구조의 예이다. 하나의 LCU는 내부에 하나 이상의 CU를 가질 수 있으며 쿼드-트리 형태로 표현 될 수 있다. 각각의 CU는 다시 예측 타입에 따라 여러 모양의 PU로 나뉘며, TU는 PU의 경계 안에서 다시 쿼드-트리 형태로 나뉘어 변환/양자화를 실시한다.

HEVC의 부호화를 수행하는 기본 유닛을 CU로 정의한다. 이것은 H.264/AVC의 매크로 블록의 개념과 유사하다. H.264/AVC의 매크로 블록과 차이점은 매크로 블록은 16x16크기의 고정 크기를 가졌지만, CU는 깊이별로 다양한 크기를 가질 수 있다는 점이다. 그림 2는 LCU의 크기가 64x64이고 계층 깊이가 4인 경우의 예

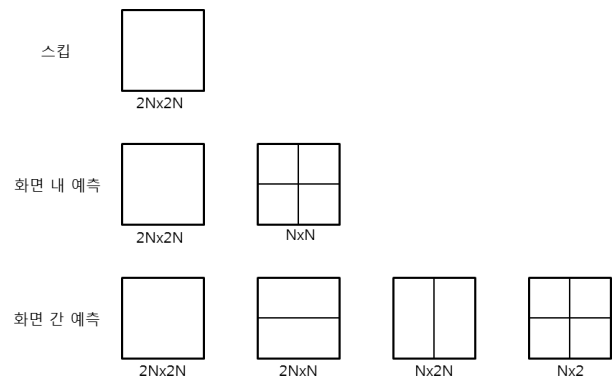


그림 3. 스킵, 화면 내 예측, 화면 간 예측을 위한 PU분할의 예  
Fig. 3. Example of possible PU splittings for skip, intra, and inter prediction.

이다. 이러한 CU의 재귀적 구조는 스플릿 플래그에 의해서 표현 가능하다. 예를 들어 깊이가 d이고 크기가 2Nx2N인 CUd에 대하여 스플릿 플래그가 0일 때에는 현재 깊이에서 CU를 부호화 하게 된다. 스플릿 플래그가 1이라면 CUd는 d+1의 깊이와 NxN 크기를 가지는 4개의 CUd+1로 나누어진다. 이 때, CUd+1는 서브-CU라 부르며 CUd+1의 깊이인 d+1이 마지막 깊이에 도달하면 CUd+1는 더 이상 나누어지지 않는다. LCU내에서 다양한 깊이를 가지는 CU들은 다시 그 내부에서 PU와 TU를 가지게 되며 각각 예측과 변환/양자화를 수행하여 부호화가 진행된다.

2Nx2N 크기의 CUd에서 예측을 수행하기 위한 기본적인 단위는 PU이다. PU는 예측 타입에 따라 그 종류가 달라진다. 그림 3과 같이 스킵은 2Nx2N PU로 수행되며 화면 내 예측 모드는 2Nx2N와 정방향으로 한번 나누어진 4개의 NxN PU로 수행된다. 화면 간 예측 모드에는 2Nx2N, 2NxN, Nx2N, NxN 크기의 PU가 예측에 사용된다. 그림 3과 같은 형태의 PU를 통해 예측을 수행한 후에 변환과 양자화를 수행하게 된다.

TU는 트랜스폼과 양자화를 수행하는 기본 단위이다. HEVC에서는 TU의 최대 크기와 최소 크기를 정의하여 사용하며, 그 크기는 최댓값과 최솟값이 SPS를 통해 전송된다. 기존의 H.264/AVC에서는 예측을 수행한 후에 8x8또는 4x4크기의 블록 단위로 트랜스폼을 수행하였으나, HEVC는 RQT (Residual Quadtree) 구조<sup>[5]</sup>를 이용하여 잔차 신호의 특성에 따라 적응적으로 TU를 나누어 부호화 하며, 이를 통해 부호화 효율을 올릴 수 있다.

그림 4는 HEVC의 CU결정 방법의 의사코드이다. 현

재 CU의 깊이에서 PU의 모양과 예측 방법에 따라 예측을 수행하며 각각 계산된 RDcost 중에서 최솟값을 가지는 모드가 현재 CU의 최적의 모드로 결정된다. 이후에 현재 CU의 깊이가 SPS에서 정의하는 최대 깊이보다 작다면 CU를 N×N크기의 서브-CU 4개로 분할할 수 있다. 분할된 N×N크기 서브-CU들의 최적의 RDcost의 합과 현재 깊이의 2N×2N크기 CU의 최적 RDcost를 비교하여 CU의 깊이와 깊이에 따른 크기를 결정한다. 이와 같은 과정은 그림 4의 의사코드와 같이 재귀적인 구조로 구현될 수 있다.

그림 4에서 출력으로 나오는 CU의 상태란, CU의 깊이, PU의 모양, 예측 모드, 움직임 벡터 등 하나의 CU를 표현하기 위한 여러 문법들이 있을 수 있다. 각 PU에 대한 RDcost는 예측을 수행하고 변환/양자화를 거쳐 재구성된 PU와 원본의 차이와 이 때 발생하는 비트를 RDcost를 계산하는 식 [6]에 의하여 계산한다. 스킵, 화면 간 예측, 화면 내 예측 방법과 각각에 해당하는 PU에 대하여 RDcost를 계산하고 이 중에서 가장 작은 값을 가지는 모드를 현재 깊이에서 최적을 모드로 선택한다. 현재 깊이에서 최적을 모드를 결정하였다면, N×N크기로 쪼개어진 4개의 서브-CU에 대하여 같은 방법으로 최적을 모드를 선택하게 되며, 이 네 개의 CU

의 RDcost의 합을 현재 깊이의 최적의 RDcost와 비교하여 최종적으로 최적의 CU를 결정한다.

HEVC의 CU결정 방법은 정의 되는 최대 깊이와 최소 깊이에 따라 CU의 크기가 최대 128×128, 최소 8×8 까지 정의될 수 있다. 또한, PU는 CU안에서 2N×2N, 2N×N, N×2N 그리고 N×N 등 다양한 크기로 예측을 수행한다. 이러한 블록 구조는 복잡한 영상과 단순한 영상 모두 효율적으로 예측할 수 있고, 이에 따라 부호화 효율을 높이는 것이 가능하다. 하지만, CU와 PU의 조합이 지나치게 많아 높은 복잡도의 주된 원인이 되고 있다. 본 논문에서는 이러한 HEVC의 높은 복잡도 문제를 해결하고자 CU결정 방법을 두 단계로 나누어 실시하여 계산 복잡도를 낮추는 알고리즘을 제안한다.

기존의 표준 코덱인 H.264/AVC의 부호화기를 위한 복잡도 감소 연구들을 살펴보면, 움직임 탐색의 방법을 개선하는 방법<sup>[7]</sup>, 주변의 문맥을 고려하여 움직임 탐색 영역을 적응적으로 하는 방법<sup>[8]</sup>, 블록 또는 움직임 벡터 등의 통계적 특성을 이용하는 방법<sup>[8-9]</sup> 등이 있으며 이 방법들은 부호화기에서 가장 복잡한 부분인 움직임 탐색을 효율적으로 수행하는 방법들이다. 이 외에도 화면 내 예측을 효율적으로 수행하는 방법에는 예측하고자 하는 블록의 특성을 고려하는 방법<sup>[10]</sup> 등이 주된 연구 방향이라 할 수 있겠다.

HEVC 부호화기의 고속화 방법 또한, 기존에 연구되었던 방법들의 연장선상에서 접근하는 것이 가능하다. 하지만 H.264는 존재하지 않던 블록 개념들이 들어가 있고 H.264의 MB (Macro Block) 개념인 CU의 결정을 고속화 하는 방법의 연구 또한 필요하다. 이에 본 논문에서도 통계적 특성에 기반한 고속 CU 결정 방법을 제안한다.

### III. 제안하는 방법

본 논문에서는 부호기의 복잡도를 감소시키기 위하여, 하나의 LCU를 부호화할 때 부호화 과정을 두 단계로 나누어 복잡도를 감소시키는 알고리즘을 제안한다. 부호화 과정의 첫 번째 단계는 2N×2N PU만을 이용하여 LCU의 깊이 정보를 획득하는 단계이며, 두 번째 단계는 첫 번째 단계에서 획득한 깊이 정보를 이용하여 해당 깊이에서만 세밀한 예측을 통해 정확한 부호화를 실시하는 단계이다. 또한, HEVC의 스킵 모드를 이용한 CU 고속 결정 방법을 함께 제안한다.

```

/*Input : depth
Output : Best CU status*/

BEGIN
    Calculate RDcost of MERGE_2N×2N
    Calculate RDcost of INTER_2N×2N
    Calculate RDcost of INTER_N×2N
    Calculate RDcost of INTER_2N×N
    IF depth == max depth THEN
        Calculate RDcost of INTER_N×N
    ENDIF
    Calculate RDcost of INTRA_2N×2N

    Select best RDcost and update its CU status

    IF current depth != max depth THEN
        FOR PartitionIndex=0 to 3 DO
            GOTO BEGIN for next depth CU with PartitionIndex
        ENDFOR
        Compare best RDcost of current depth with accumulated RDcost of
        the next depth and update its CU status
    ENDIF
END
    
```

그림 4. HM3.0의 CU 결정 의사코드  
Fig. 4. Pseudo code of CU decision for HM3.0.

1. 깊이 우선 결정을 통한 고속 CU 결정 방법

본 논문은 첫 번째 단계에서 낮은 복잡도로 깊이 정보를 예측하기 위해서 PU 선택에 관한 통계적 특성을 이용하였다. 실험 영상을 통해 2N×2N PU의 비율을 측정하면 표 1과 같으며, 대부분의 영상에서 2N×2N PU의 선택이 절대적으로 높은 통계적 특성을 확인할 수 있다. 이러한 통계적 특성을 이용하여 제안하는 방법의 첫 번째 단계에서는 2N×2N 크기의 PU만을 사용하여 예측을 수행하고, RD (Rate Distortion) 경쟁을 통해 LCU 내의 깊이를 결정 한다. 2N×2N PU만을 이용하면 낮은 복잡도로 깊이를 결정하는 것이 가능하고 비교적 정확한 깊이 정보를 획득하는 것이 가능하다.

그림 5는 제안하는 알고리즘의 첫 번째 단계를 나타낸 의사코드이다. 첫 번째 단계에서는 현재 프레임 내의 부호화 하고자 하는 2N×2N CU가 입력으로 들어왔을 때, LCU 내의 각 CU에 대한 깊이 정보를 구하는 것을 목적으로 한다. 여기서 깊이 정보란 하나의 LCU 안에서 CU의 쪼개짐을 나타내는 쿼드-트리 형태를 의미한다. 첫 번째 단계에서는 2N×2N 스킵, 2N×2N 화면 내 예측, 그리고 2N×2N 화면 간 예측을 수행한다. 예

표 1. 2N×2N PU 선택 비율  
Table 1. 2N×2N PU selection ratios.

Class	영상	2N×2N PU 선택 비율 (%)	
		Random Access	Low Delay_P
A	Traffic	81.5	-
	PeopleOnStreet	85.1	-
	NebutaFestival	91.9	-
	SteamLocomotive	90.9	-
B	Kimono	80.9	81.3
	ParkScene	74.5	77.4
	Cactus	82.0	84.1
	BasketballDrive	71.7	73.7
	BQTerrace	79.1	81.2
C	BasketballDrill	84.7	87.3
	BQMall	86.1	84.4
	PartyScene	75.1	74.8
	RaceHorses	78.3	79.9
D	BasketballPass	85.2	84.2
	BQSquare	77.5	75.9
	BlowingBubbles	79.1	76.8
	RaceHorses	73.1	76.0
E	Vidyo1	-	81.7
	Vidyo3	-	79.9
	Vidyo4	-	75.3
평균		80.98	79.62

측을 수행할 때, 2N×2N PU만을 사용하는 것이 특징이며 예측을 수행한 후의 부호화 과정은 기존의 HM3.3과 동일하다. 예측을 수행한 후에 현재 CU의 깊이가 3보다 작아, 서브-CU로 쪼개어 질 수 있다면 4개의 서브-CU로 쪼개어 깊이를 증가시키며 그림 4에서 설명한 것과 같은 동작을 재귀적으로 수행한다. 또한, 현재 깊이가 3 즉, 8×8 크기의 CU라면 더 이상 서브-CU로 쪼개어지지 않는다. 첫 번째 단계는 가장 많이 선택되는 2N×2N PU만을 부호화 하여 낮은 복잡도로 CU 깊이 정보를 구하였고, 2N×2N PU의 선택비율이 지배적이기 때문에, 적은 복잡도로 효과적으로 깊이 정보를 획득할 수 있다.

첫 번째 단계에서 획득한 CU 깊이 정보는 두 번째 단계의 입력으로 사용된다. 그림 6은 두 번째 단계의 의사코드이며, 두 번째 단계에서는 첫 번째 단계에서 계산한 CU 깊이 정보를 이용하여 현재 2N×2N CU의 깊이가 첫 번째 단계에서 구한 CU의 깊이와 동일할 때에만 부호화를 수행하게 된다. 즉, 첫 번째 단계에서 구한 CU의 깊이와 현재의 깊이가 같지 않다면, CU는 다시 네 개의 서브-CU로 나누어진다. 서브-CU의 깊이는 원 CU의 깊이에 1을 더한 깊이가 되며, 이 과정은 현재 CU가 첫 번째 단계에서 구한 CU의 깊이와 동일할 때까지 재귀적으로 수행된다.

현재 CU의 깊이가 첫 번째 단계에서 구한 CU의 깊이와 동일하다면, 예측을 수행하는데 이때에는 2N×2N,

```

/*Input : depth
Output : CU depth information*/

BEGIN
    Calculate RDcost of MERGE_2N×2N
    Calculate RDcost of INTER_2N×2N
    Calculate RDcost of INTRA_2N×2N

    Select best RDcost and update its CU status

    IF current depth != max depth THEN
        FOR PartitionIndex=0 to 3 DO
            GOTO BEGIN for next depth CU with PartitionIndex
        ENDFOR
        Compare best RDcost of current depth with accumulated RDcost of the next depth and update its CU status
    ENDFIF
END
    
```

그림 5. 제안하는 첫 번째 과정의 의사코드  
Fig. 5. Pseudo code for the first stage of the proposed method.

```

/*Input : depth, CU depth information
Output : Best CU status*/

BEGIN
  IF current depth == decided depth information at the first stage THEN
    Calculate RDcost of MERGE_2N×2N
    IF RDcost of the MERGE_2N×2N > cumulative average of the Merge
    SKIP RDcost && all cbf==0 THEN
      GOTO END
    ENDIF
    Calculate RDcost of INTER_2N×2N
    Calculate RDcost of INTER_N×2N
    Calculate RDcost of INTER_2N×N
    IF depth == max depth THEN
      Calculate RDcost of INTER_N×N
    ENDIF
    Calculate RDcost of INTRA_2N×2N

    Select best RDcost and update its CU status
  ELSE
    FOR PartitionIndex=0 to 3 DO
      GOTO BEGIN for next depth CU with PartitionIndex
    ENDFOR
    Compare best RDcost of current depth with accumulated RDcost of
    the next depth and update its CU status
  ENDIF
END
    
```

그림 6. 제안하는 두 번째 과정의 의사코드  
 Fig. 6. Pseudo code for the second stage of the proposed method.

2N×N, N×2N, N×N 크기의 가능한 모든 PU에 대한 예측을 수행하게 된다. 예외적으로 N×N 크기의 PU의 경우 CU의 깊이가 마지막 깊이일 때에만 수행한다. 두 번째 단계는 모든 CU 깊이에서 예측을 수행하지 않고, 첫 번째 단계에서 정해진 깊이에서만 예측을 수행하기 때문에 첫 번째 단계의 깊이 정보가 정확하다면 부호화 효율의 감소 없이 효율적인 예측이 가능하다.

2. 스킵을 이용한 CU 조기 종료 방법

LCU를 부호화 하는 과정을 두 단계로 나누어 복잡도를 줄이는 방법과 더불어 본 논문에서는 복잡도 감소 효과의 증대를 위해서 스킵을 이용한 CU 조기 종료 방법을 제안한다. HEVC의 스킵은 머지 스킵이<sup>[11]</sup> 사용된다. 머지 스킵은 주변 블록 중 참조하는 블록의 인덱스를 보내며, 잔차신호는 선택적으로 전송한다. 이 인덱스를 머지 인덱스라 하며, 인덱스를 참조하는 블록의 움직임 정보를 통해 부호화 하고자 하는 블록의 움직임 정보를 알 수 있다. 따라서, 주변 PU와 부호화 하고자

표 2. 스킵 모드 선택 비율  
 Table 2. Skip mode selection ratios.

Class	영상	스킵모드 선택 비율 (%)	
		Random Access	Low Delay_P
A	Traffic	22.5	-
	PeopleOnStreet	51.6	-
	NebutaFestival	63.9	-
	SteamLocomotive	73.0	-
B	Kimono	77.1	77.1
	ParkScene	59.1	59.1
	Cactus	61.3	61.3
	BasketballDrive	38.1	38.1
	BQTerrace	49.1	49.1
C	BasketballDrill	52.3	41.8
	BQMall	59.1	54.9
	PartyScene	45.1	41.7
	RaceHorses	33.2	31.4
D	BasketballPass	53.3	58.4
	BQSquare	65.2	55.7
	BlowingBubbles	55.2	46.8
	RaceHorses	43.9	36.7
E	Vidyo1	-	81.7
	Vidyo3	-	79.9
	Vidyo4	-	75.3
평균		53.12	55.56

하는 PU의 움직임이 비슷하다면 스킵모드로 부호화 되어, 움직임 벡터와 잔차신호의 생략이 가능하며 이에 따라 높은 부호화 효율을 얻을 수 있다. 표 2는 HEVC의 실험 영상에<sup>[12]</sup> 대한 스킵 모드 선택 비율을 나타낸 것이다.

표 2에서 살펴볼 수 있듯이 보통의 자연 영상에서는 스킵 모드 선택 비율이 일반적으로 높게 나타나는 것을 알 수 있다. 이러한 통계적 특성을 바탕으로 본 논문에서는 빠른 CU 결정을 위한 방법 중 하나로 스킵을 이용한 조기 CU 결정 방법을 제안한다.

제안하는 방법은 식 (1)의 조건과 같이 스킵의 RDcost가 일정 문턱치를 넘지 않고, 잔차신호를 변환 및 양자화하여 나온 양자화 계수가 존재하지 않는다면, 현재 CU의 나머지 부호화 과정을 생략한다.

$$RDcost_{n,m,SKIP} < \frac{TH \times accumRDcost_{n,m,SKIP}}{N} \quad (1)$$

여기서,  $RDcost_{n,m,SKIP}$ 은 스킵의 RDcost를 의미하며,  $n, m$ 은 각각 CU의 넓이와 높이를 의미한다. 또한,  $TH$ 는 제한하는 방법의 문턱치로 실험적으로 결정하는 값이다.  $accumRDcost_{n,m,SKIP}$ 는 너비와 높이가 각각  $n$  그

리고  $m$ 인 CU에서 스킵에 대한 RDcost를 의미하며, 이 값은 해당 깊이에서 최적의 값을 가지는 경우에만 누적 수행한 결과이다. 마지막으로  $N$ 은 RDcost가 누적된 횟수를 의미한다.

스킵을 이용한 조기 CU 결정 방법은 부호화 과정을 두 단계로 나누어 수행하는 방법을 결합하여 더욱 효과적인 복잡도 감소 효과를 얻을 수 있다.

#### IV. 실험 결과

제안하는 방법의 검증을 위해서, 본 논문은 현재 표준화 과정에서 실험을 위해 작성한 공통 실험조건 문서 [12]를 참조하였으며, Random Access, Low Delay P 조건에서 각 영상별로 150 프레임씩 실험을 진행하였다. 복잡도와 압축 성능의 검증을 위해서 제안하는 방법과 HM 3.3을 비교하여 BD-bitrate와 시간 절감율을 계산하였으며, 시간 절감율은 식 (2)와 같이 계산하였다.

$$\Delta T = \frac{Time_{ref} - Time_{pro}}{Time_{ref}} \quad (2)$$

식 (2)에서  $Time_{ref}$ 는 HM3.3의 부호화 시간이며,  $Time_{pro}$ 는 본 논문에서 제안하는 방법의 부호화 시간이다. 스킵을 이용한 조기 CU 결정을 위한 문턱치  $TH$ 는 실험적으로 1.5를 사용하였다.

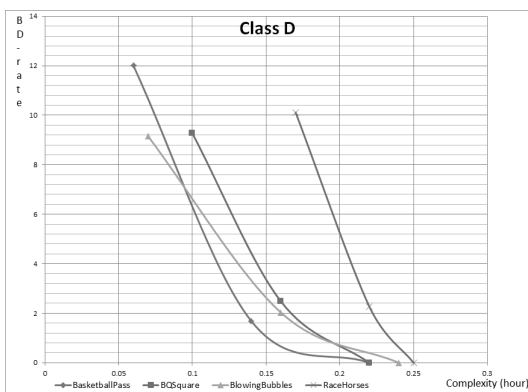


그림 7. 조기 종료율을 위한 문턱치에 따른 복잡도, BD-rate 그래프

Fig. 7. Complexity and BD-rate graph by early skip threshold.

표 3. 실험 조건  
Table 3. Test condition

CPU	Intel(R) Coer(TM) i7
RAM	4.00GB
Parallel processing 여부	사용하지 않음

그림 7은 실험 영상에서 class D 4개 영상으로  $TH$ 를 구하기 위한 실험을 진행한 결과이다. 하나의 그래프는 각각 하나의 영상을 나타내고 있다. 그래프에서 왼쪽 점은  $TH$ 를 2.0으로 실험한 결과이며, 가운데 점은  $TH$ 가 1.5이고 오른쪽 점은 수정하지 않은 HM3.3의 결과이다. 그림 7을 보면 일반적으로  $TH$ 가 1.5인 점 부근에서 BD-rate가 급격하게 증가하는 것을 알 수 있고, 증가하는 비트율 대비 복잡도 감소량이 많은 지점인 것을 확인할 수 있다.

실험 환경은 표 3에 정리되어 있다. 고속화에 적용된 방법은 알고리즘 개선을 통한 방법만을 사용하였으며, 기타의 병렬화 방법들은 배제하였다.

HM 3.3과 제안하는 방법의 RD 성능과 시간 절감율을 표 4, 5, 6 그리고 7에 각각 나타내었다. 표 4는 random access high efficiency 환경에서 제안하는 방법과 HM 3.3을 비교하여 계산한 BD-bitrate와 시간 절감율을 나타내고 있다. 전체적으로 제안하는 방법은 1.82%의 비트율 증가와 함께, 약 50% 시간 절감율을 보인다. 표 4의 random access 환경은 30프레임 주기로 한 장의 I 프레임이 삽입되어 있으며, HM 3.3에서 I 프레임은 8×8크기의 CU를 제외하면 2N×2N PU만을 사용한다. 결과적으로 제안하는 방법에서 I 프레임은 시간 절감 효과를 거의 얻지 못하게 된다. 따라서 표 4에서 나타난 시간 절감율은 대부분 B 프레임들 통한 것이 되며, B 프레임 안에서 CU 결정 과정을 효과적으로 간소화한 결과이다. 비교하는 HM 3.3은 모든 깊이에서 모든 크기의 PU를 사용함으로써, 영상 특성에 맞는 가장 좋은 깊이의 CU와 CU 안에서 그 블록의 특성에 맞는 PU를 선택할 수 있기 때문에 부호화 효율이 높다. 하지만 CU와 PU를 조합하는 경우의 수가 너무 많기 때문에, 제안하는 방법에 비하여 복잡도가 높은 것을 확인할 수 있다. 제안하는 방법은 2N×2N PU가 통계적으로 가장 많이 나온다는 특성을 이용하여 첫 번째 깊이 결정 단계에서는 모든 크기의 PU를 사용하지 않기 때문에, 기존의 방법 대비 복잡도 절감 효과를 확인할 수 있다. 또한, CU의 깊이 결정 단계에서 결정된 깊이에서는 다른 종류의 PU에 대한 부호화를 실시하여 복잡도와 부호화 효율간의 트레이드 오프를 고려한 결과를 보이고 있다.

제안하는 방법은 영상의 크기와 특성에 따라 결과가 달리 나오는 것을 확인할 수 있는데, 표 4에서 특히 높은 비트 증가율을 보이는 class A의 Traffic영상은 국

표 4. Random access high efficiency 환경에서 제안한 방법의 BDBitrate와 시간 절감율

Table 4. BDBitrates and time saving of the proposed method in random access high efficiency configuration

Class	영상	QP	HM 3.3			제안한 방법			BD-rate	ΔT (%)
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)		
A	Traffic	22	13203.2	41.85	5.52	13155.36	41.78	2.69	2.66	51.33
		27	5381.4	39.30	5.01	5362.59	39.20	1.90		62.01
		32	2548.8	36.72	4.80	2540.24	36.61	1.58		67.01
		37	1294.6	34.07	4.67	1287.85	33.96	1.39		70.25
	PeopleOnStreet	22	33482.3	40.45	7.07	33637.36	40.42	5.10	2.09	27.95
		27	16102.5	37.43	6.22	16212.09	37.38	3.91		37.09
		32	8401.6	34.45	5.75	8460.09	34.37	3.18		44.66
	NebutaFestival	22	224457.0	39.58	9.62	224636.39	39.58	7.37	0.97	23.43
		27	98790.3	33.71	8.03	98854.92	33.70	6.02		25.04
		32	32114.7	29.77	6.22	31889.27	29.73	3.13		49.66
		37	8411.6	28.03	5.30	8371.03	27.97	1.73		67.26
	SteamLocomotive	22	37255.8	40.61	6.33	37081.56	40.58	3.44	1.14	45.62
27		10045.6	38.65	5.38	9983.19	38.62	2.02	62.50		
32		4311.9	37.12	5.01	4292.42	37.08	1.62	67.62		
B	Kimono	27	2019.3	35.13	4.82	2000.13	35.09	1.38	1.61	71.46
		32	5614.5	42.08	3.71	5619.8	42.06	2.13		42.78
		37	2644.1	40.10	3.26	2646.3	40.06	1.57		51.96
	ParkScene	22	1292.6	37.64	3.03	1292.3	37.57	1.20	2.19	60.31
		27	644.2	35.13	2.75	642.6	35.06	0.94		65.69
		32	787.6	40.12	4.02	785.7	40.06	2.01		50.06
		37	3394.2	37.53	3.48	3390.7	37.46	1.37		60.67
	Cactus	22	1528.6	34.90	3.11	1525.0	34.82	1.08	1.65	65.20
		27	688.4	32.38	3.05	686.4	32.31	0.92		69.84
		32	1990.87	38.58	3.49	19867.3	38.56	2.05		41.35
		37	6549.0	36.91	2.80	6559.1	36.88	1.25		55.51
		22	3053.6	34.98	2.59	3059.3	34.94	1.02		60.78
37		1521.0	32.79	2.51	1522.1	32.75	0.89	64.67		
Basketball Drive	22	17415.0	39.45	4.64	17483.4	39.43	2.48	1.67	46.61	
	27	6195.1	37.81	3.81	6222.4	37.79	1.72		54.92	
	32	2884.1	35.99	3.40	2902.6	35.96	1.39		59.25	
	37	1473.9	34.01	3.33	1478.2	33.97	1.21		63.79	
	22	38505.7	38.08	4.53	38406.7	38.05	2.77		39.01	
	37	9153.8	35.76	3.45	9008.5	35.72	1.42		58.93	
Basketball Drive	22	3272.3	34.11	3.05	3250.8	34.07	1.04	1.54	65.78	
	27	1454.6	32.21	3.01	1446.2	32.18	0.90		70.09	
	32	3577.0	40.72	0.68	3592.8	40.68	0.42		38.72	
	37	1722.6	37.58	0.58	1729.4	37.53	0.32		44.92	
	22	831.5	34.68	0.53	835.1	34.63	0.26		51.35	
	37	421.6	32.17	0.50	422.8	32.10	0.19		62.00	
C	BQMall	22	5146.9	39.72	0.66	5174.6	39.69	0.43	1.97	35.33
		27	2464.4	37.02	0.57	2479.5	36.97	0.33		41.41
		32	1240.3	34.13	0.52	1249.7	34.06	0.24		53.07
	PartyScene	22	642.4	31.31	0.59	644.2	31.24	0.20	1.76	65.70
		27	8608.9	38.27	0.94	8624.6	38.22	0.62		34.21
		32	4068.2	34.56	0.82	4077.0	34.50	0.45		45.29
		37	1964.5	31.23	0.68	1964.2	31.15	0.35		48.23
		22	937.1	28.19	0.59	935.6	28.10	0.26		56.24
		37	6218.1	38.37	1.11	6244.8	38.35	0.78		29.62
	RaceHorses	22	2538.2	34.80	0.92	2552.9	34.75	0.59	2.36	35.76
		27	1150.2	31.90	0.74	1156.4	31.81	0.40		45.53
		37	523.2	29.14	0.61	524.6	29.04	0.27		55.17
D	Basketball Pass	22	1165.0	41.35	0.20	1169.9	41.31	0.11	1.45	45.38
		27	583.5	37.63	0.17	587.6	37.59	0.09		44.52
		32	286.6	34.28	0.16	288.3	34.24	0.08		52.02
	BQSquare	22	144.2	31.44	0.15	144.8	31.39	0.06	1.66	58.48
		27	1784.8	38.72	0.20	1782.3	38.63	0.13		38.39
		32	775.1	35.22	0.17	772.0	35.12	0.09		50.32
		37	398.4	32.27	0.16	396.4	32.18	0.07		57.31
	BlowingBubbles	22	220.1	29.41	0.15	218.1	29.31	0.06	2.08	55.49
		27	1538.9	38.61	0.22	1544.2	38.55	0.14		36.86
		32	704.0	35.39	0.18	705.2	35.31	0.10		45.26
		37	332.8	32.30	0.17	332.6	32.22	0.08		55.70
	RaceHorses	22	155.9	29.49	0.16	156.5	29.43	0.06	2.46	61.05
27		1334.6	39.11	0.23	1344.6	39.06	0.20	14.42		
32		655.6	35.31	0.20	660.1	35.23	0.15	23.64		
37		311.8	31.77	0.18	313.4	31.67	0.11	38.87		
평균									1.82	50.18

표 5. Random access low complexity 환경에서 제안한 방법의 BDBitrate와 시간 절감율

Table 5. BDBitrates and time saving of the proposed method in random access low complexity configuration.

Class	Seq.	QP	HM 3.3			제안한 방법			BD-rate	ΔT (%)
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)		
A	Traffic	22	13907.5	41.55	5.00	13801.1	41.46	1.91	3.00	61.80
		27	5682.5	39.03	4.43	5659.2	38.92	1.27		71.39
		32	2713.6	36.49	4.20	2706.3	36.37	0.97		76.80
		37	1393.2	33.84	3.89	1389.6	33.73	0.81		79.29
	PeopleOnStreet	22	34820.5	40.07	5.73	34966.8	40.02	3.89	2.31	32.03
		27	16997.6	37.12	5.28	17140.6	37.06	3.01		42.95
		32	8959.1	34.16	4.90	9063.6	34.09	2.31		52.94
		37	5010.4	31.42	4.69	5078.0	31.33	1.87		60.06
	NebutaFestival	22	238328.3	38.73	7.31	238542.2	38.72	4.94	1.41	32.39
		27	111951.0	33.20	6.34	112359.6	33.19	4.19		33.99
		32	29869.3	29.25	5.08	29674.9	29.19	1.99		60.87
		37	8646.1	27.83	4.40	8616.3	27.77	1.06		75.87
SteamLocomotive	22	36944.6	40.09	5.13	36761.6	40.06	2.42	2.19	52.85	
	27	10431.6	38.36	4.45	10436.3	38.33	1.34		69.93	
	32	4546.1	36.86	4.16	4587.6	36.83	1.01		75.67	
	37	2215.8	34.89	4.01	2217.1	34.86	0.83		79.20	
Kimono	22	9041.7	39.91	4.61	9043.3	39.85	3.20	2.34	30.44	
	27	3593.5	36.97	3.79	3602.8	36.91	2.08		45.21	
	32	1503.7	34.17	3.43	1509.3	34.10	1.53		55.45	
	37	633.9	31.60	3.24	634.2	31.53	1.23		62.23	
	22	22832.8	38.65	4.70	22869.7	38.62	3.42		27.29	
	37	6716.5	36.65	3.64	6749.5	36.62	2.02		44.44	
ParkScene	22	2923.8	34.52	3.23	2945.5	34.47	1.52	2.34	52.84	
	27	1379.1	32.23	3.05	1384.8	32.17	1.27		58.49	
	32	54504.4	38.89	5.48	54824.2	38.87	4.51		17.73	
	37	10697.3	35.61	3.92	10608.5	35.54	2.21		43.60	
Cactus	22	2950.7	33.66	3.35	2920.8	33.61	1.39	2.40	58.57	
	27	1020.6	31.60	3.14	999.8	31.54	1.09		65.24	
	32	20170.6	39.65	5.22	20225.4	39.64	3.68		29.51	
	37	7085.0	37.84	4.28	7113.0	37.82	2.50		41.55	
	22	3273.8	35.94	3.80	3298.0	35.91	1.88		50.55	
	37	1623.8	33.83	3.53	1634.1	33.79	1.54		56.47	
Basketball Drive	22	6547.9	42.33	4.86	6560.6	42.32	3.31	1.61	31.89	
	27	3171.4	40.38	4.23	3179.6	40.35	2.52		40.28	
	32	1541.0	37.67	3.63	1543.0	37.64	1.96		45.92	
	37	741.0	34.90	3.25	745.0	34.87	1.58		51.33	
	22	3754.8	40.34	0.79	3763.9	40.30	0.43		44.95	
	37	1810.4	37.24	0.71	1814.8	37.19	0.32		55.02	
Basketball Drill	22	877.4	34.39	0.66	879.7	34.33	0.24	1.69	63.74	
	27	447.3	31.95	0.63	448.5	31.87	0.19		70.30	
	32	5470.0	39.51	0.76	5497.6	39.47	0.43		43.35	
	37	2604.1	36.84	0.69	2622.5	36.78	0.33		51.86	
	22	1318.7	33.98	0.64	1331.3	33.91	0.26		59.27	
	37	692.1	31.19	0.61	700.1	31.12	0.21		65.16	
PartyScene	22	9021.0	37.95	0.85	9034.4	37.90	0.53	2.31	37.10	
	27	4250.7	34.30	0.75	4251.3	34.22	0.38		50.02	
	32	2046.0	31.06	0.69	2041.5	30.95	0.27		60.24	
	37	977.2	28.06	0.65	972.8	27.96	0.20		68.43	
	22	6423.7	38.03	1.02	6448.3	38.00	0.71		29.86	
	37	2634.5	34.63	0.88	2651.6	34.57	0.51		42.47	
RaceHorses	22	1208.1	31.77	0.79	1212.6	31.67	0.37	2.62	53.27	
	27	551.1	29.05	0.73	553.3	28.93	0.28		61.81	
	32	1217								



표 6. Low delay high efficiency 환경에서 제안한 방법의 BDbitrate와 시간 절감율

Table 6. BDbitrates and time saving of the proposed method in low delay high efficiency configuration.

Class	Seq.	HM 3.3			제안한 방법			BD-rate	ΔT (%)
		QP	Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)		
B	Kimono	22	7013.0	41.98	3.99	7013.0	41.78	2.85	28.59
		27	3236.8	39.99	3.59	3236.7	39.96	2.28	36.36
		32	1490.7	37.52	3.26	1490.7	37.51	1.88	42.54
		37	762.3	35.00	2.77	766.3	34.99	1.19	57.06
	ParkScene	22	9041.7	39.91	4.71	9061.9	39.89	2.20	35.30
		27	3593.5	36.97	3.92	3609.1	36.91	1.37	53.46
		32	1503.7	34.17	3.53	1513.7	34.11	1.01	62.35
	Cactus	27	633.9	31.60	3.34	638.9	31.58	0.73	71.56
		22	22832.8	38.65	4.83	22862.8	38.61	2.18	36.17
		27	6716.5	36.65	3.77	6746.2	36.59	1.42	49.55
	Basketball Drive	32	2923.8	34.52	3.30	2933.1	34.50	1.08	58.02
		37	1379.1	32.23	3.10	1386.1	32.19	0.86	64.68
22		20170.6	39.65	5.18	20170.6	39.62	2.55	34.33	
BQTerrace	27	7085.0	37.84	4.26	7085.0	37.80	1.80	46.17	
	32	3273.8	35.94	3.77	3273.8	35.89	1.36	54.37	
	37	1623.8	33.83	3.50	1623.8	33.78	1.09	60.85	
C	Basketball Drill	22	54504.4	38.89	5.56	54504.4	38.83	2.61	32.60
		27	10697.3	35.61	3.99	10697.3	35.58	1.39	52.74
		32	2950.7	33.66	3.41	2950.7	33.61	0.92	64.03
	BQMall	37	1020.6	31.60	3.18	1020.6	31.55	0.73	69.83
		22	4261.9	39.92	0.49	4272.0	39.89	0.35	29.82
		27	1958.5	36.74	0.43	1964.3	36.69	0.25	41.29
	PartyScene	32	914.7	33.98	0.39	923.0	33.92	0.19	52.42
		37	446.0	31.53	0.36	449.7	31.46	0.14	61.13
		22	6575.8	39.38	0.49	6619.6	39.35	0.35	28.07
	RaceHorses	32	2905.5	36.33	0.42	2930.7	36.28	0.25	39.72
		37	1374.1	33.32	0.38	1391.4	33.26	0.19	48.83
		22	672.8	30.42	0.35	680.5	30.35	0.15	56.37
Basketball Pass	22	12119.3	37.83	0.59	12154.8	37.79	0.50	15.57	
	27	5030.0	33.82	0.48	5007.5	33.73	0.32	33.55	
	32	2164.4	30.42	0.42	2163.5	30.32	0.22	46.58	
BQSquare	37	916.0	27.25	0.38	914.7	27.13	0.16	57.48	
	22	8212.7	38.96	0.62	8248.9	38.94	0.56	8.89	
	27	3119.5	34.84	0.50	3125.9	34.78	0.40	21.45	
BlowingBubbles	32	1315.7	31.62	0.43	1326.0	31.56	0.29	34.26	
	37	545.1	28.75	0.39	550.4	28.67	0.21	45.42	
	22	1392.6	40.86	0.10	1399.0	40.82	0.08	22.07	
RaceHorses	27	682.5	37.13	0.09	687.1	37.09	0.06	31.84	
	32	329.0	33.80	0.08	331.6	33.74	0.05	41.80	
	37	160.8	30.94	0.08	162.5	30.91	0.04	50.15	
BQSquare	22	3009.8	38.25	0.12	3039.0	38.18	0.10	9.59	
	27	1082.5	33.95	0.09	1063.4	33.82	0.06	31.33	
	32	429.4	30.89	0.08	423.1	30.75	0.04	48.35	
RaceHorses	37	178.1	27.90	0.07	176.2	27.77	0.03	59.05	
	22	1932.7	38.02	0.11	1936.6	37.95	0.10	12.05	
	27	781.6	34.49	0.09	781.3	34.39	0.07	30.14	
BlowingBubbles	32	329.9	31.34	0.08	331.5	31.26	0.05	44.41	
	37	142.9	28.52	0.08	143.5	28.45	0.03	54.32	
	22	1498.0	40.44	0.11	1499.2	40.41	0.07	36.31	
RaceHorses	27	701.5	35.46	0.09	704.0	35.34	0.06	38.95	
	32	302.3	31.72	0.08	306.3	31.64	0.04	46.74	
	37	141.5	28.90	0.08	141.0	28.86	0.03	55.78	
E	Vidyo1	22	1868.1	43.74	1.08	1870.9	43.70	0.53	51.15
		27	655.8	41.54	1.07	656.9	41.48	0.40	62.49
		32	308.9	39.12	1.00	309.2	39.05	0.39	61.39
		37	159.3	36.39	0.98	160.8	36.34	0.35	63.74
	Vidyo3	22	3017.3	43.44	1.11	3020.7	43.40	0.58	47.56
		27	1017.7	41.05	1.09	1025.5	40.98	0.43	60.89
		32	475.4	38.46	0.99	481.0	38.38	0.39	61.18
		37	249.5	35.51	0.97	252.0	35.44	0.33	65.58
	Vidyo4	22	3227.0	42.93	1.17	3233.3	42.90	0.67	42.98
		27	1021.7	40.58	1.14	1026.8	40.54	0.49	56.59
		32	443.7	38.16	1.03	444.9	38.11	0.42	58.84
		37	221.7	35.58	0.96	222.7	35.54	0.32	66.53
평균								2.15	45.99

표 7. Low delay low complexity 환경에서 제안한 방법의 BDbitrate와 시간 절감율

Table 7. BDbitrates and time saving of the proposed method in low delay low complexity configuration.

Class	Seq.	HM 3.3			제안한 방법			BD-rate	ΔT (%)
		QP	Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)		
B	Kimono	22	7211.0	41.86	3.69	7229.4	41.84	2.32	37.16
		27	3436.8	39.89	3.29	3458.8	39.87	1.80	45.29
		32	1670.7	37.32	2.98	1684.6	37.28	1.37	54.01
		37	819.3	34.70	2.76	826.7	34.63	1.04	62.53
	ParkScene	22	9554.2	39.68	3.40	9540.6	39.60	2.09	38.57
		27	3801.4	36.79	2.93	3812.6	36.72	1.36	53.65
		32	1601.1	34.04	2.69	1610.5	33.97	0.96	64.42
	Cactus	37	680.5	31.49	2.56	684.7	31.42	0.72	71.85
		22	23980.4	38.43	3.42	24012.8	38.39	2.18	36.16
		27	7229.9	36.47	2.82	7275.3	36.44	1.37	51.52
	Basketball Drive	32	3173.1	34.36	2.57	3202.6	34.32	1.02	60.11
		37	1514.1	32.08	2.44	1531.0	32.03	0.82	66.50
22		21610.8	39.38	3.88	21618.7	39.35	2.52	35.05	
BQTerrace	27	7764.7	37.56	3.34	7794.3	37.53	1.74	47.84	
	32	3599.9	35.62	2.98	3628.6	35.59	1.31	56.23	
	37	1797.7	33.54	2.78	1814.5	33.50	1.03	62.85	
C	Basketball Drill	22	54277.9	38.37	3.88	54319.7	38.31	2.89	25.43
		27	11457.6	35.34	2.94	11282.6	35.25	1.36	53.79
		32	3262.6	33.35	2.56	3231.3	33.29	0.84	67.29
	BQMall	37	1146.9	31.20	2.41	1129.2	31.14	0.62	74.43
		22	4028.1	40.27	1.14	4034.7	40.24	0.87	26.49
		27	1851.9	37.11	0.96	1860.5	37.07	0.62	41.22
	PartyScene	32	865.1	34.31	0.83	869.6	34.25	0.45	53.06
		37	419.4	31.79	0.76	420.8	31.70	0.34	59.60
		22	6140.4	39.63	1.13	6181.6	39.61	0.87	25.32
	RaceHorses	27	2731.9	36.56	0.92	2753.6	36.52	0.61	33.55
		32	1292.4	33.52	0.81	1309.8	33.46	0.47	42.41
		37	630.7	30.63	0.74	637.2	30.55	0.38	57.02
Basketball Pass	22	11668.0	38.07	1.44	11701.0	38.04	1.27	19.04	
	27	4849.6	33.96	1.10	4829.1	33.88	0.81	31.22	
	32	2095.3	30.52	0.92	2093.5	30.42	0.56	44.90	
BQSquare	37	877.1	27.31	0.80	877.6	27.21	0.40	53.60	
	22	8064.2	39.55	1.59	8107.6	39.54	1.40	29.71	
	27	3043.8	35.11	1.24	3053.1	35.07	0.96	34.37	
BlowingBubbles	32	1263.7	31.79	1.03	1274.6	31.73	0.67	45.44	
	37	519.8	28.85	0.90	523.9	28.78	0.50	50.80	
	22	1331.3	41.20	0.26	1335.2	41.17	0.20	24.51	
PartyScene	27	650.8	37.39	0.23	657.0	37.35	0.15	33.71	
	32	313.4	33.97	0.20	315.5	33.92	0.12	41.31	
	37	152.0	31.05	0.18	152.6	30.98	0.09	48.13	
RaceHorses	22	2913.4	38.56	0.32	2931.0	38.51	0.28	21.00	
	27	1059.7	34.13	0.24	1039.1	33.99	0.17	30.08	
	32	422.6	30.97	0.20	415.6	30.83	0.11	45.11	
BlowingBubbles	37	173.8	27.91	0.18	170.1	27.74	0.08	54.50	
	22	1878.7	38.22	0.31	1880.4	38.16	0.26	24.86	
	27	757.0	34.60	0.23	758.8	34.51	0.17	38.68	
BQSquare	32	318.1	31.38	0.19	318.9	31.30	0.12	51.72	
	37	135.3	28.51	0.17	136.3	28.45	0.09	63.88	
	22	1548.0	39.44	0.32	1557.2	39.41	0.33	24.23	
RaceHorses	27	723.5	35.29	0.26	730.0	35.24	0.24	41.06	
	32	332.3	31.61	0.22	335.3	31.54	0.17	59.42	
	37	149.5	28.80	0.19	151.0	28.70	0.13	61.66	
Vidyo1	22	1985.5	43.26	1.05	1972.4	43.20	0.37	64.90	
	27	695.5	41.10	0.98	699.3	41.05	0.27	72.85	
	32	328.5	38.73	0.94	331.6	38.68	0.23	76.04	
	37	176.3	36.09	0.92	178.0	36.06	0.21	77.70	
Vidyo3	22	3309.1	42.87	1.07	3310.3	42.82	0.42	60.32	
	2								

소적인 움직임이 많을 뿐만 아니라 영상이 전체적으로 빠르게 움직이는 특성을 가지고 있다. 반면 class A의 NebutaFestival 영상은 다소 복잡하지만 원래의 영상 크기가 2560×1600으로 크고 영상의 전체적인 움직임이 비교적 작아 효과적인 복잡도 감소와 함께 비트 증가율 또한, 적은 것을 확인할 수 있다.

표 5는 random access low complexity 환경에서 제안하는 방법과 HM 3.3을 비교하여 계산한 BD-bitrate와 시간 절감율을 나타내고 있다. 표 4의 random access high efficiency 환경과 비교할 때 random access low complexity 환경에서는 복잡도가 높은 루프 필터인 ALF (Adaptive Loop Filter)를 사용하지 않으며, 엔트로피 코더로는 복잡도가 높은 CABAC (Context Adaptive Binary Arithmetic Coding) 대신 상대적으로 복잡도가 낮은 CAVLC (Context Adaptive Variable Length Coding)를 사용한다. 따라서 random access low complexity 환경에서는 제안하는 방법에서 고속화하려는 CU 결정 단계가 부호화기에서 차지하는 부분이 상대적으로 크다. 이것은 부호화기에서 제안하는 방법이 고속화하려는 부분의 비율이 크다는 것을 의미하며, low complexity 환경이 high efficiency 환경보다 복잡도 감소 효과가 클 수 있다는 것을 의미하기도 한다. 하지만, random access high efficiency가 잘못된 깊이 예측에 따른 화질 열화를 감소시켜 줄 수 있는 루프 필터를 취하는 반면, random access low complexity는 그 화질 열화를 대부분 그대로 가지고 있기 때문에, 복잡도 감소 효과는 약 52%로 더 크지만 비트 증가율 또한, 약 0.2% 더 높은 것을 확인할 수 있다.

Low delay high efficiency와 Low delay low complexity 환경에서 제안하는 방법과 HM 3.3을 비교하여 계산한 BD-bitrate와 시간 절감율은 표 6 그리고 7에 각각 나타나 있다. Random access 환경에서의 결과와 비교했을 때, 비트는 약 0.1~0.3% 증가하였고, 시간 절감율은 약 2~5% 감소한 결과를 확인할 수 있다. 공통 실험조건 문서<sup>[12]</sup>를 참조하면, low delay는 실험 영상으로 2560×1600 크기의 class A 영상을 사용하지 않는다. 표 1 그리고 표 2를 보면 class A 영상은 2N×2N PU의 선택 비율이 높고, 스킵 비율 또한 다른 영상에 비하여 높은 것을 확인할 수 있다. 따라서 class A가 포함되지 않은 low delay 환경은 random access와 비교하여 상대적으로 낮은 성능을 보인다. 또한, low delay 환경은 GOP (Group of Picture) 구조가 IPPP..형

태이며, random access와 같이 중간에 삽입되는 I 프레임이 존재하지 않는다. 따라서 잘못된 깊이 선택에 따른 화질 열화가 I 프레임을 부호화한 뒤 사라지는 random access 환경과 달리 뒤쪽 프레임에 계속해서 영향을 줄 수 있으며, 이에 따라 비트 증가율이 random access 환경에서의 결과와 비교하여 다소 높은 경향을 나타낸다.

Low delay low complexity 환경은 low delay high efficiency 환경과 비교하였을 때, ALF, CABAC 등의 복잡도 높은 툴을 사용하지 않아 전체 부호화기에서 제안하는 방법 적용 비율이 높다. 따라서 복잡도 감소 효과도 상대적으로 높은 것을 확인할 수 있다. 하지만, 화질 열화를 감소시킬 수 있는 루프 필터를 적용하지 않기 때문에 상대적으로 높은 비트 증가율을 보이고 있다.

실험 영상으로 사용된 영상은 영상의 복잡함과 움직임 정도에 따라 분류될 수 있으며, 적용된 알고리즘이 각 영상의 미치는 영향을 살펴 볼 수 있다. 표 4, 5, 6 그리고 7을 살펴보면 class A의 Traffic, class B의 Cactus, class C의 RaceHorsesC 그리고 class D의 RaceHorses 등을 영상은 다른 영상과 비교하였을 때 상대적으로 복잡한 영상과 부분적 혹은 전체적으로 빠른 움직임을 가지는 영상들이다. 이와 같은 영상들은 타 영상에 비하여 BD-rate가 높게 측정되었다.

## V. 결 론

본 논문에서는 LCU의 부호화 단계에서 PU의 크기에 따른 통계적 특성을 이용하여, LCU 부호화를 두 단계로 나누었다. 첫 번째 단계는 깊이 정보를 획득하는 단계이고, 두 번째 단계는 제한된 깊이에서 PU와 TU를 결정하는 단계이다. 제안하는 방법을 HM 3.3과 비교 실험한 결과 심각한 화질의 열화 없이 약 50%의 부호화 시간 감소효과를 얻을 수 있었다.

이후 보다 낮은 복잡도로 정확한 깊이 정보를 획득할 수 있는 방법에 대하여 연구할 계획이다.

## 참 고 문 헌

- [1] ISO/IEC JTC1/SC29/WG11, "Vision, Applications and Requirements for High-performance Video Coding (HVC)," N11096, Jan 2010.
- [2] Thomas Wiegand, Gary J. Sullivan, Gisle

- Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," IEEE trans. on Circuits and Systems for Video Technology, vol. 13, no.7, pp. 560-576, July 2003.
- [3] T. Wiegand, B. Bross, W.J. Han, J.R. ohm, G.J. sullivan, "High Efficiency Video Coding (HEVC) text specification Working Draft 1", JCTVC-C403, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Guangzhou, CN, Oct. 2010.
- [4] T. Wiegand, B. Bross, W.J. Han, J.R. ohm, G.J. sullivan, "WD3: Working Draft 3 of High-Efficiency Video Coding", JCTVC-E603, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Geneva, CH, Mar. 2011.
- [5] T. Wiegand, H. Schwarz B. Bross, A. Fuldseth, X. Wang, W.J. Han, "BoG report: residual quadtree structure", JCTVC-C319, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Guangzhou, CN, Oct. 2010.
- [6] K. McCann, S. Sekiguci, B. Bross, W.J. Han, "HEVC Test Model 3 (HM 3) Encoder Description", JCTVC-E603, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Geneva, CH, Mar. 2011.
- [7] 유중훈, 오승준, 안창범, 박호중, "적응형 윈도우 크기 기반 NTSS," 전자공학회논문지, 제43권 SP편, 제1호, 75-84쪽, 2006년 1월.
- [8] 이제운, 최웅일, 전병우, 석민수, "H.264의 가변 블록 움직임 보상을 위한 고속 움직임 벡터 탐색 및 모드 결정법," 전자공학회논문지, 제40권 SP편, 제4호, 49-59쪽, 2003년 7월.
- [9] H.Q. Zeng, C.H. Cai, K.-K. Ma "Fast mode decision for H.264/AVC based on macroblock motion activity," IEEE transactions on circuits and system for video technology, vol. 19, no. 4, Apr. 2009.
- [10] 임웅, 남정학, 정광수, 심동규, "통계적 확률 기반의 효율적인 고속 화면 내 예측 방법," 전자공학회 논문지, 제47권 SP편, 제3호, 44-53쪽, 2010년 5월.
- [11] B. Bross, J. Jung, Y.-W. Huang, Y.H. Tan, Il-Koo Kim, T. Sugio, M. Zhou, T.K. Tan, E. Francois, Kimihiko Kazui, Wei-Jung Chien, S. Sekiguchi, S. Park, Wade Wan, "BoG report of CE9: Mv Coding and Skip/Merge operations", JCTVC-E481, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Geneva, CH, Mar. 2011.
- [12] Frank Bossen, "Common test conditions and software reference configurations," JCTVC-E700, Joint Collaborative team on Video Coding of ITU-T SG16 Wp3 and ISO/IEC JTC1/SC29/WG11, Geneva, CH, Mar. 2011.

---

 저 자 소 개
 

---



유성은(학생회원)  
 2011년 광운대학교 컴퓨터공학과  
 학사 졸업.  
 2011년~현재 광운대학교 컴퓨터  
 공학과 석사과정.  
 <주관심분야 : 영상신호처리, 영  
 상 압축>



안용조(학생회원)  
 2010년 광운대학교 컴퓨터공학과  
 학사 졸업.  
 2012년 광운대학교 컴퓨터공학과  
 석사 졸업.  
 2012년~현재 광운대학교 컴퓨터  
 공학과 박사과정.  
 <주관심분야 : 영상 압축, 멀티프로세서>



심동규(정회원)  
 1999년 서강대학교 전자공학과  
 공학박사.  
 1999년~2000년 (주) 현대 전자.  
 2000년~2002년 (주) 바로 비전.  
 2002년~2005년 Univ. of  
 Washington

2005년~현재 광운대학교 컴퓨터공학과 (부교수)  
 <주관심분야 : 영상신호처리, 영상압축, 컴퓨터  
 비전>