

논문 2016-53-11-6

움직임 벡터에 의한 관심영역 기반의 HEVC 고속 부호화 유닛 결정 방법

(Fast Coding Unit Decision Algorithm Based on Region of Interest by Motion Vector in HEVC)

황 인 서*, 선우 명 훈**

(In Seo Hwang and Myung Hoon Sunwoo[©])

요 약

고효율 영상 부호화 기술인 high efficiency video coding (HEVC)은 부호화 효율을 높이기 위하여 coding tree unit (CTU)을 사용한다. CTU는 coding unit (CU), prediction unit (PU), transform unit (TU)으로 구성되며 모든 가능한 경우의 CU, PU, TU 분할연산을 통해 최적의 분할 조합을 찾아내게 된다. 블록 분할 연산의 복잡도를 감소시키기 위하여 본 논문은 움직임 벡터에 의한 관심 영역 CTU 추출에 근거하는 PU 분할 결정 방법과 이전에 부호화된 프레임의 같은 위치의 CTU 정보를 사용하는 CU 깊이 결정 분할 알고리즘을 제안한다. 첫 번째 방법은 프레임 중 움직임이 많은 동적 CTU 부분과 움직임이 적은 정적 CTU 부분으로 나누어 정적인 영역에 대해 PU 분할 연산을 감소시키는 방법이며, 두 번째 방법은 이전 프레임의 CTU 깊이 정보를 기반으로 현재 CTU의 분할 깊이를 미리 예측하여 CU 분할 연산을 감소시킨다. 결과적으로 제안하는 알고리즘은 HEVC test model (HM) 14.0 버전 대비 BDBR 손실은 2.5% 발생했지만, 전체 부호화 시간이 약 44.8%로 크게 감소했다.

Abstract

High efficiency video coding (HEVC) employs a coding tree unit (CTU) to improve the coding efficiency. A CTU consists of coding units (CU), prediction units (PU), and transform units (TU). All possible block partitions should be performed on each depth level to obtain the best combination of CUs, PUs, and TUs. To reduce the complexity of block partitioning process, this paper proposes the PU mode skip algorithm with region of interest (RoI) selection using motion vector. In addition, this paper presents the CU depth level skip algorithm using the co-located block information in the previously encoded frames. First, the RoI selection algorithm distinguishes between dynamic CTUs and static CTUs and then, asymmetric motion partitioning (AMP) blocks are skipped in the static CTUs. Second, the depth level skip algorithm predicts the most probable target depth level from average depth in one CTU. The experimental results show that the proposed fast CU decision algorithm can reduce the total encoding time up to 44.8% compared to the HEVC test model (HM) 14.0 reference software encoder. Moreover, the proposed algorithm shows only 2.5% Bjontegaard delta bit rate (BDBR) loss.

Keywords : high efficiency video coding (HEVC), HEVC test model (HM) 14.0, region of interest (RoI), coding unit (CU), prediction unit (PU)

* 정회원, LIG 넥스원 (LIG Nex1)

** 평생회원, 아주대학교 전자공학과 (Department of Electrical and Computer Engineering, Ajou University)

© Corresponding Author (E-mail : sunwoo@ajou.ac.kr)

※ 본 논문은 미래창조과학부 중견연구자지원사업[2014R1A2A2A01002952, 미래형 실감미디어 지원을 위한 영상처리 전용 프로세서 연구]의 지원에 의하여 연구되었음.

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업 (IITP-2016-R2718-16-0015)의 결과로 수행되었음

I. 서론

디지털 TV, 다양한 모바일 기기에서의 실시간 초고화질 영상 재생 서비스의 수요가 빠르게 증가함에 따라서 고화질 영상 전송을 위한 높은 성능의 압축 기술은 필수적이다. 이러한 요구를 충족시키기 위하여 ITU-T video coding experts group (VCEG) 과 ISO/IEC moving picture experts group (MPEG) 은 joint collaborative team on video coding (JCT-VC) 을 결성하여 새로운 영상 압축 표준인 high efficiency video coding (HEVC) 을 개발하였다^[1~3]. 고해상도 영상에 대응하기 위하여 HEVC 는 context adaptive binary arithmetic coding (CABAC) 방식을 채택하였으며 이전 세대의 표준과 달리 쿼드트리 방식의 계층적 부호화 개념을 도입하였다^[1]. 결과적으로 기존의 영상 압축 표준인 H.264/AVC 와 비교했을 때 같은 영상 화질을 유지하면서 50% 정도의 비트를 감소를 달성하였다^[3~4].

기존의 HEVC 표준에 따르면 하나의 CTU당 모든 CU 깊이에 대해 가능한 모든 PU와 TU 조합의 윌-왜곡 비용을 계산한다. 최상위 CU 깊이부터 최하위 깊이까지 존재할 수 있는 모든 경우의 윌-왜곡 비용을 비교하여 가장 낮은 비용을 가지는 조합으로 CTU의 부호화를 결정한다. HEVC는 우수한 압축 효율을 보이지만 존재하는 모든 경우의 수에 대한 조합의 윌-왜곡 비용을 계산하기 때문에 엄청난 연산 복잡도를 가진다. 고해상도 영상으로 갈수록 처리해야 하는 데이터의 양이 점점 많아져 최근의 멀티미디어 사용 경향인 고해상도 멀티미디어 영상의 실시간 재생 목적에 부합하지 않는 실정이다.

고속 부호화 효율을 달성하기 위하여 각 CTU 별로 부호화할 CU 깊이를 적응적으로 결정하여 불필요한 깊이에서의 윌-왜곡 비용 연산을 생략하는 형태의 고속 알고리즘을 제안한다. 적응적 CU 깊이에 기반하여 부호화 CU 깊이를 제한하는 여러 알고리즘에 따르면^[10~12] 기존 알고리즘은 주변의 CTU가 가지는 CU 깊이 정도에 따라 현재 부호화할 CTU의 CU 깊이의 범위를 문턱 값으로 한정해 그 값을 넘어가는 깊이에 대해 연산을 생략하는 형태의 알고리즘이다. 이 방법은 단순히 공간적으로 주변에 있는 CTU의 CU 분할 정보를 바탕으로 문턱 값 계산을 하는 다소 단순한 방법으로 수행되는 반면, 제안하는 알고리즘은 시간상으로 이전 프레임의 동일 위치에 있는 CTU 정보를 이용하는 시간적 중복 특성을 활용하는 방법이다. 본 논문에서는 이전 프레임 CTU의 움직임 벡터값을 이용한다. 움직임이 많은 동적 CTU인 관심영역과 그렇지 않은 정적 CTU 영역을 비관심영역으로 구분하고 시간적으로 이전에 부호화된 CTU가 가지는 값이 평균 정보를 조합하여 경우에 따라 많은 양의 윌-왜곡 비용 연산을 감소시킬 수 있는 고속 부호화 알고리즘을 제안한다. 2장에서는 기존 HEVC 표준이 채택하고 있는 재귀적 쿼드트리 구조 및 제안하는 알고리즘에 대한 설명, 3장에서는 제안하는 알고리즘에 대한 실험결과 및 분석, 그리고 마지막 4장에서 결론을 맺는다.

II. 본론

1. HEVC 표준 부호화 알고리즘

최신 영상 부호화 기술인 HEVC는 프레임의 부호화

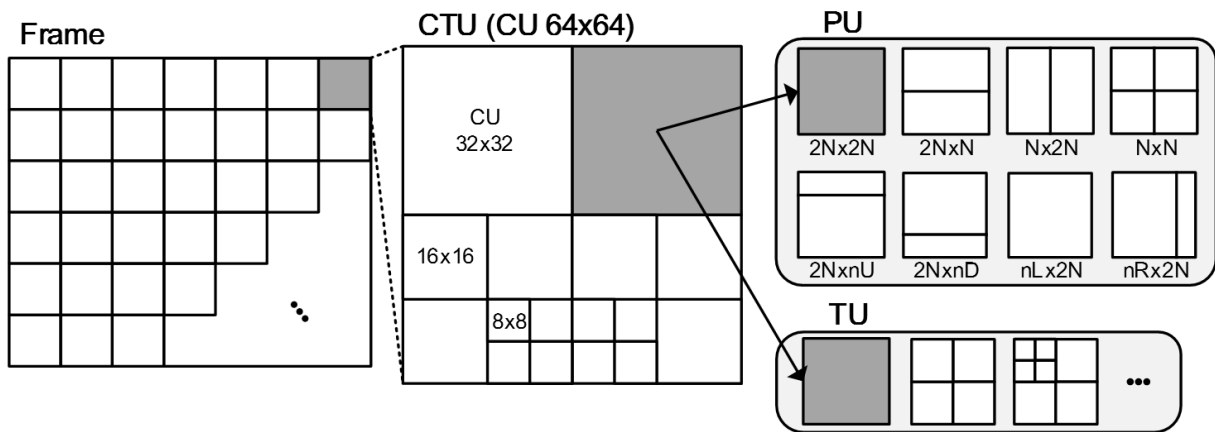


그림 1. HEVC 부호화 유닛 조합 구성도
Fig. 1. HEVC encoding block combination of CU, PU and TU.

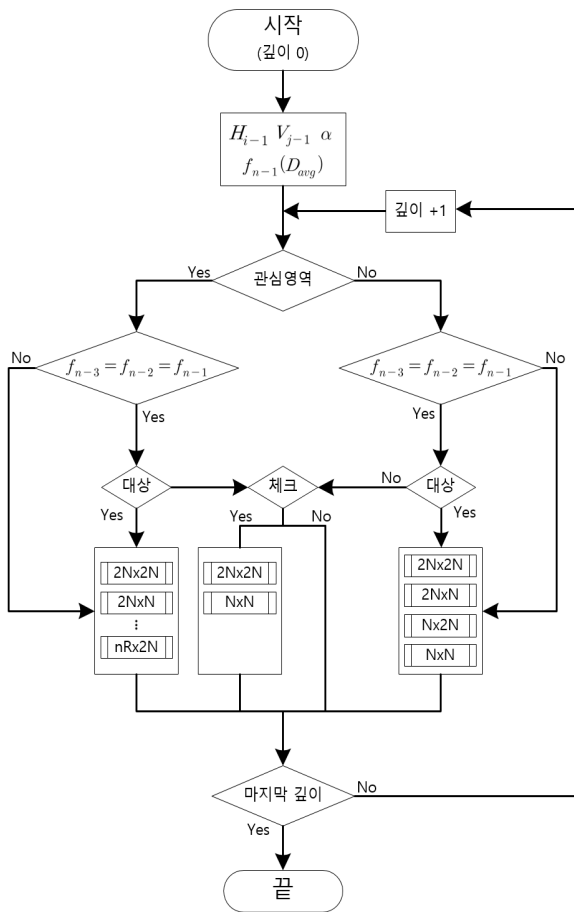


그림 2. 제안하는 알고리즘 순서도
Fig. 2. Flow chart of proposed algorithm.

가 이루어지는 기본 단위로 CTU를 사용하며 이것은 이전 표준인 H.264/AVC에서 사용하는 매크로 블록과 비슷한 개념이다. 그림 1은 영상의 프레임이 CTU로 나뉘고 그에 대한 CU, PU 및 TU의 조합이 어떻게 존재할 수 있는지를 보여주는 예이다. 하나의 CTU를 부호화하기 위하여 그 내부의 CU 깊이를 결정하게 되는데 최상위 크기 64x64의 깊이 0부터 시작해서 깊이가 하나씩 증가할 때마다 CU가 4분할로 나누어지는 과정이 재귀적으로 반복되어 크기가 8x8인 최종 깊이 3까지 탐색을 하게 된다. CU의 깊이를 탐색하는 과정 중 깊이별로 PU와 TU의 모든 조합에 대한 율-왜곡 비용을 계산하여 가장 낮은 비용을 가지는 CU의 깊이 및 PU와 TU의 조합이 정해지게 되고 이것은 영상의 프레임에 존재하는 모든 CTU에 대하여 동일하게 적용된다.

2. 제안하는 알고리즘

전체 부호화 과정의 고속화를 위해 본 논문에서는 영상의 움직임 벡터 값을 바탕으로 관심영역 CTU를 결

표 1. 변수 정의

Table 1. Definition of Variables.

기호	정의
H_k	k 번째 행의 움직임 벡터 크기 평균
V_k	k 번째 열의 움직임 벡터 크기 평균
$\alpha(H_{i-1})$	프레임 내 CTU 행의 $i-1$ 번째 문턱 값
$\alpha(V_{j-1})$	프레임 내 CTU 열의 $j-1$ 번째 문턱 값
f_{n-k}	현재 프레임 기준 시간적으로 k 번째 전 프레임
γ	탐색 영역의 10% 값 ($64 \times 0.1 \approx 6$)
k_i	깊이 배열 i 번째 위치의 깊이 k 값
D_{avg}	하나의 CTU가 가지는 깊이 평균 값

정하고 관심영역과 비관심영역에 대해 다른 연산을 적용하는 하는 고속 부호화 알고리즘을 제안한다. 이를 위한 두 가지의 방법으로 PU 모드 조기 결정 방법 및 CU 깊이 생략 방법을 설계하였다. 표 1에서 사용한 변수에 대한 정의를 정리하였고 그림 2에서 제안하는 알고리즘의 전체적인 흐름을 순서도로 나타내었다.

가. 방법1: 움직임 벡터기반 PU 모드 조기 결정

방법1은 시간상으로 이전 프레임에서 추출된 관심영역 CTU를 기반으로 한다. CTU 내부의 CU 깊이와 PU 예측 모드는 물체의 움직임에 따른 움직임 복잡성과 카메라의 움직임에 따른 움직임 균일성^[9] 등의 정보를 바탕으로 결정된다. 시간상으로 이전의 프레임과 현재의 프레임 간의 값 변화가 있으므로 이 차이가 블록 단위의 움직임 벡터 값으로 나타나게 된다.

모든 영상의 프레임은 CTU 블록 단위로 나뉘게 되므로 하나의 CTU 당 내부의 움직임 벡터 값들을 모두 모아 블록 내부의 움직임 정도를 파악하면 상대적으로 움직임이 많은 블록과 그렇지 않은 블록으로 나눌 수 있을 것이다. 이것을 동적 CTU, 정적 CTU 라고 각각 정의하고 제안하는 알고리즘에서는 움직임이 많은 영역인 동적 CTU를 관심영역으로 설정한다.

관심영역인 동적 CTU 블록은 이전 프레임의 값과 비교했을 때 비관심영역에 비해 상대적으로 값이 많이 다르므로 더욱 정확한 값의 예측을 위해 기존에 HEVC에서 정의하는 8개의 PU 모드를 모두 사용해 예측 연산을 수행한다. 관심영역으로 판단되지 않은 블록은 값의 변화가 거의 없는 영상의 배경 부분이거나 움직임이 적은 정적 CTU이다. 이러한 영역에 대해서는 PU 모드를 모두 사용해 자세히 예측을 수행 하는 것이 아닌 비대칭 PU 모드 ($2N \times nU$, $2N \times nD$, $nL \times 2N$, $nR \times 2N$) 예측 단계를 생략해 빠르게 PU 예측 연산을 종료한다.

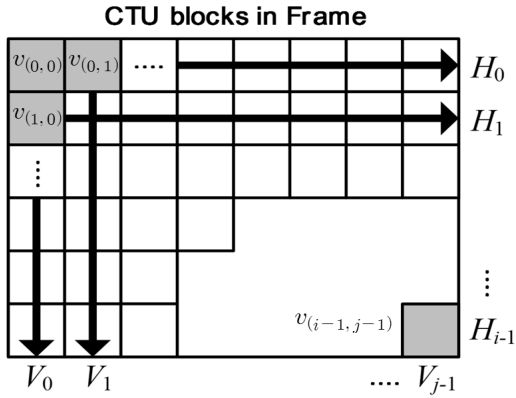


그림 3. 수평, 수직방향 행, 열 평균값 연산
Fig. 3. Row and column values of horizontal and vertical direction.

(1) 프레임의 움직임 벡터 행, 열 평균 연산

그림 3에 나와 있는 대로 하나의 CTU 내부에 존재하는 움직임 벡터 값을 모두 누적해 대표 움직임 벡터를 (\$v_{(0,0)} \sim v_{(i-1, j-1)}\$) CTU의 개수만큼 구한다. 대표 벡터의 크기를 구한 뒤 식 (1)을 사용하여 수평 방향으로 행 평균값을 (\$H_0 \sim H_{i-1}\$) 구하고 식 (2)를 사용해 수직 방향으로 열 평균값을 (\$V_0 \sim V_{j-1}\$) 구한다.

$$H_{i-1} = \frac{1}{j} \sum_{k=0}^{j-1} |v_{(i-1, k)}| \quad (1)$$

$$V_{j-1} = \frac{1}{i} \sum_{k=0}^{i-1} |v_{(k, j-1)}| \quad (2)$$

(2) 관심영역 추출을 위한 문턱 값 \$\alpha\$ 연산

현재 프레임의 관심영역을 추출을 위한 문턱 값 \$\alpha\$를 결정하기 위해 시간상으로 이전 프레임에서 연산된 평균값을 사용한다. 식 (3)과 (4)에 따라 프레임에 존재하는 행, 열 개수만큼의 \$\alpha\$값을 결정하기 위해 시간적으로 이전의 첫 번째, 두 번째 프레임 인덱스에 해당하는 행, 열 평균값을 이용하여 각각 0.5의 가중치를 곱해 \$\alpha\$를 결정하게 된다. \$\gamma\$값은 동적 CTU 추출과정에서 행, 열 평균과 비슷한 대표 움직임 벡터 크기 값을 가지는 CTU가 관심영역으로 추출되는 것을 제한하고 벡터의 크기가 평균보다 확실히 큰 값을 가지는 CTU에 대해서만 관심영역으로 추출하기 위한 오프셋 값이며 탐색 영역의 10% 값을 (\$64*0.1 \approx 6\$) 할당해주었다.

$$\alpha(H_{i-1}) = 0.5 \times f_{n-2}(H_{i-1}) + 0.5 \times f_{n-1}(H_{i-1}) + \gamma \quad (3)$$

$$\alpha(V_{j-1}) = 0.5 \times f_{n-2}(V_{j-1}) + 0.5 \times f_{n-1}(V_{j-1}) + \gamma \quad (4)$$

식 (5)를 사용하여 두 개의 정해진 문턱 값과 CTU 별 대표 움직임 벡터의 크기를 비교하여 크기가 두 문턱 값과 비교해 하나라도 크면 해당 CTU를 관심영역으로 추출한다.

$$\alpha(H_{i-1}) < |v_{(i-1, j-1)}| \quad (5)$$

$$\alpha(V_{j-1}) < |v_{(i-1, j-1)}|$$

나. 방법2: 적응적 CU 깊이 생략

영상의 프레임은 일반적으로 공간적, 시간적 중복성의 특징을 가지게 된다. 제한하는 방식은 이러한 특징을 활용한 것으로 현재의 CTU가 부호화 결과로 가지게 될 CU의 깊이는 이전에 먼저 부호화 된 CTU와 비슷한 값을 가진다는 결과에 기반을 두고 있다^[7]. 시간상으로 이전 프레임의 동일 위치에 있는 CTU의 깊이 정보를 이용하는 시간적 중복 특성을 활용하는 방법을 제안한다.

(1) CTU 내부의 CU 깊이 평균

일반적인 영상은 인접한 프레임들끼리의 시간적 중복 현상이 두드러지게 나타난다. 따라서 현재 프레임의 CTU의 CU 깊이는 이전 프레임에서 같은 위치에 해당하는 CTU의 CU 깊이 정보와 같거나 매우 비슷한 값을 가질 확률이 높으므로 이전에 부호화가 완료된 CTU의 정보를 활용하여 현재 부호화될 CTU의 CU 깊이 범위를 예상하여 해당 범위를 벗어나는 깊이를 생략하는 방법으로 연산을 빠르게 수행할 수 있다.

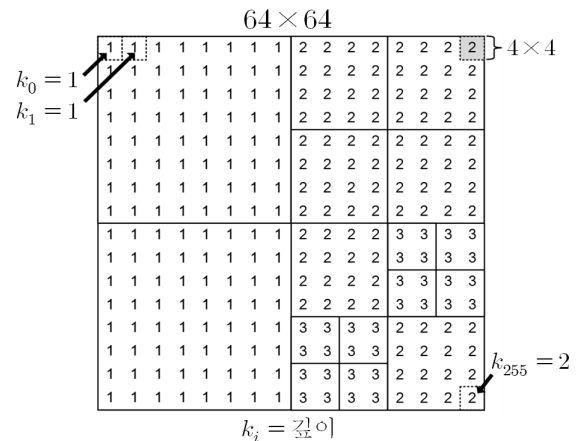


그림 4. CTU 내부의 CU 깊이 저장 배열
Fig. 4. CU depth array in CTU.

HM 14.0 기준, 모든 CTU는 256개의 CU 깊이 저장 공간을 가지고 있다. 그림 4에서 하나의 CTU가 다양한 깊이의 CU로 나뉘고 그 깊이 배열에 저장되는 예시가

표 2. 깊이 평균값에 따른 대상 깊이, 체크깊이 설정
Table2. Setting the target depth and check depth based on average depth.

깊이 평균	대상 깊이	체크 깊이	생략 깊이
$D_{avg} = 0$	0	1	2, 3
$0 < D_{avg} \leq 1$	1	0, 2	3
$1 < D_{avg} \leq 2.5$	1, 2, 3	-	0
$2.5 < D_{avg} \leq 3$	2, 3	1	0

나타나 있다. CU 깊이 저장 배열로부터 식 (6)에 따라 CTU 별 깊이 평균을 구할 수 있게 되고 이 값을 바탕으로 현재 부호화 될 CTU의 CU 깊이를 예측할 수 있다.

$$D_{avg} = \frac{1}{256} \sum_{i=0}^{255} k_i \quad (6)$$

(2) 깊이 평균값에 의한 대상 깊이, 체크 깊이의 설정
현재 프레임의 CTU를 부호화하기 위해 프레임에서 동일한 위치에 존재하는 과거 3개의 CTU 정보를 확인하여 주변보다 시간적 중복성이 두드러지게 나타나는 CTU에 대해서 불필요한 CU 깊이 연산을 생략한다. 즉, 시간상으로 다른 이전 3개의 프레임에 대해 (f_{n-3} , f_{n-2} , f_{n-1}) 동일한 위치에 있는 CTU 3개의 깊이 평균값 각각이 모두 표 2에서 설정한 범위중 하나에 모두 속하는 경우를 고려한다. 현재 프레임의 동일위치에 존재하는 CTU의 부호화를 진행하는 과정에서 이전 3개의 CTU 깊이 평균이 속한 범위에 해당하는 대상 깊이, 체크 깊이에 대해서만 PU 예측 연산을 실시하고 생략 깊이에 대해서는 연산을 생략하는 방법을 사용한다. 기존 HEVC 표준이 모든 CU 깊이에 대한 PU 예측을 하는 것에 비해 연산을 빠르게 종료시킬 수 있다.

첫 번째, 대상 깊이란 프레임 간 시간적 중복성 특징이 두드러지는 깊이로 현재 CTU가 부호화되었을 때 이전 CTU의 깊이 평균값 근처에서 현재 깊이가 결정될 확률이 매우 높기 때문에 이 깊이에 대해서는 모든 PU 예측 연산을 자세하게 수행한다. 두 번째, 체크 깊이란 이전 프레임과 중복성이 크지 않아 영상에서 화면이 갑자기 변화하는 등의 일부 경우에 따라 낮은 확률로 이전의 깊이 평균과 비교적 많이 다르게 나누어질 수 있는 경우로 이 깊이에 대해서는 2Nx2N, NxN 의 2가지 경우의 PU에 대해서만 빠르게 예측 연산을 실시하고 나머지 6개의 경우에 대해서는 실시하지 않는다.

세 번째, 생략 깊이에서는 모든 PU에 대해 예측 연산을 하지 않는다. 즉, 해당 CU 깊이에서의 생략이 이루어지게 된다.

지난 프레임 3개의 CTU가 가지는 평균값이 모두 같은 범위에 들어가지 않는 경우에 대해서는 대상 깊이, 체크 깊이, 생략깊이를 설정하지 않고 모든 깊이에 대해서 방법1만 사용하여 예측연산을 수행한다.

표 3. 방법의 조합에 따른 PU 블록 예측 정도
Table3. Case of PU blocks for method1, 2 combination.

방법1 \ 방법2	$f_{n-3} = f_{n-2} = f_{n-1}$			방법2 적용x
	대상 깊이	체크 깊이	생략 깊이	
관심영역 (동적 CTU)	2Nx2N			2Nx2N
	2NxN			2NxN
	Nx2N			Nx2N
	NxN	2Nx2N		NxN
	2NxnU	NxN	-	2NxnU
	2NxnD			2NxnD
	nLx2N nRx2N			nLx2N nRx2N
비관심영역 (정적 CTU)	2Nx2N			2Nx2N
	2NxN	2Nx2N		2NxN
	Nx2N	2NxN		Nx2N
	NxN			NxN

제안하는 방법1, 2 간의 우선순위는 방법1로 설정한다. 모든 가능한 경우의 예측 연산을 표 3에 정리하였다.

표 4. 시뮬레이션 환경
Table4. Simulation environment.

소프트웨어	HM 14.0
운영체제	Windows 7
CPU	Intel core i7 @ 3.40GHz
메모리	16GB DRAM
양자화 매개변수	22, 27, 32, 37

III. 실험

표 4에 명시한 시스템 환경에서 시뮬레이션을 수행하였다. HM과 제안하는 알고리즘의 성능비교를 위해 영상별 BDBR 값과 부호화 시간을 계산하였다.

$$TS(\%) = \frac{T_{ref} - T_{prop}}{T_{ref}} \times 100 \quad (7)$$

T_{ref} 는 HM의 부호화 시간이고 T_{prop} 는 제안하는 알고리즘의 부호화 시간으로 식 (7)에 근거하여 HM 대

표 5. 제안하는 알고리즘의 시뮬레이션 결과

Table5. Simulation results of proposed algorithm.

		Proposed		[7]		[8]		[9]	
		BDBR (%)	TS (%)	BDBR (%)	TS (%)	BDBR (%)	TS (%)	BDBR (%)	TS (%)
1920x1080	BQTerrace	0.7	55.6	1.3	42.7	0.3	21.5	0.3	47.5
	Cactus	1.6	43.0	0.5	43.5	0.3	19.2	2.2	48.0
	Kimono	1.9	43.6	0.8	38.0	0.5	26.5	0.8	47.9
1280x720	FourPeople	1.9	50.7	1.1	64.0	0.0	16.2	1.6	65.6
832x480	PartyScene	2.6	40.5	1.0	40.0	0.2	13.9	0.4	33.0
	RaceHorsesC	3.4	40.7	1.0	34.1	0.4	15.9	1.1	26.1
	BasketballDrill	2.0	42.0	1.9	44.0	0.1	14.2	2.0	37.0
416x240	RaceHorses	4.7	44.9	0.7	30.8	0.4	39.9	0.7	21.2
	BasketballPass	3.6	43.2	1.5	35.4	0.1	12.7	1.2	27.7
	BlowingBubbles	3.0	44.1	2.2	39.7	0.3	12.7	0.3	31.5
Average		2.5	44.8	1.2	41.2	0.3	19.3	1.1	38.6

비 본 알고리즘의 시간 감소율을 백분율로 계산하였다. 양자화 매개변수를 달리해서 시뮬레이션 한 4개의 측정값의 BDBR과 TS값을 구한 뒤 평균값을 구하여 표 5에 결과를 입력 하였다. 영상 각각의 평균값에 대해 다시 평균을 계산하여 전체 평균값을 계산해 알고리즘의 전체 성능을 구하여 타 논문과의 성능 및 영상별 시뮬레이션 결과 비교가 원활히 이루어질 수 있도록 하였다. 실험 결과로 HM 14.0 대비 최대 55.6%, 최소 40.5% 평균 약 44.8%의 부호화 시간 감소를 이루어 냈다. 타 논문들과 비교해볼 때 [7]에서 최대 64.0%, 최소 30.8%, [8]에서 최대 39.9%, 최소 12.7% [9]에서 최대 65.6%, 최소 21.2%의 성능을 가진다. 제안하는 알고리즘의 시간감소율의 최대, 최소 편차가 15.1%이며 타 논문의 편차인 33.2%^[7], 27.2%^[8], 44.4%^[9]에 비해 다소 낮은 수치로 고속 부호화 기 성능의 안정적인 동작을 보증한다.

측정한 BDBR 값에 대해서는 평균적으로 2.5%의 증가율을 보이는데 낮은 해상도의 416x240 WQVGA 크기의 영상에서 4.7%까지 나왔지만 해상도가 커질수록 BDBR 수치가 점점 좋아져 가장 큰 해상도인 1920x1080 Full-HD 사이즈에서 최대 0.7%까지 성능이 향상되었다. 본 알고리즘의 특성상 정적인 부분이 많은 영상에서 성능이 좋게 측정되기 때문에 고해상도 영상이 가지는 특징 중 넓은 배경영역으로 인한 정적인 부분이 많이 존재하는 성질로 해상도가 높은 영상의 BDBR 성능이 타 논문과 비교했을 때 좋아지는 것을 확인하였다.

IV. 결 론

제안하는 알고리즘은 CTU 별로 내부에 존재하는 움직임 벡터 값을 사용해 문턱 값을 설정한 뒤 상대적으

로 움직임이 많은 영역에 대해 관심영역으로 추출하는 방법1과 프레임 간 시간적 중복성에 기반하는 CTU 별 평균 CU 깊이를 사용해 적응적 깊이 예측을 하는 방법 2를 사용하였다. WQVGA 해상도부터 FHD에 이르기까지 최소 40% 이상, 평균 약 45%의 부호화 시간 감소율 달성으로 다양한 해상도의 실시간 고속 부호화 수요에 적절히 대응할 수 있다. BDBR 손실의 평균 2.5% 달성으로 최근에 늘어나는 HD (1280x720)급 이상의 고해상도 영상의 부호화 수요에 적용하기에 적절한 성능을 보여준다.

REFERENCES

- [1] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, Y.-K. Wang and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Consent)," document JCTVC-L1003 of JCT-VC, Geneva, Switzerland, Jan. 2013.
- [2] 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, Dec. 2012.
- [3] Y. Ahn, T. Hwang, S. Yoo, W. Han, D. Sim, "Statistical Characteristics and Complexity Analysis of HEVC Encoder Software," Journal of Broadcast Engineering, vol. 17, no. 6, pp. 1091-1105, Nov. 2012.
- [4] M.T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?," IEEE Consumer Electron. Mag., vol. 1, no. 3, pp. 36-46, Jul. 2012.
- [5] G. Zhong, X. He, L. Qing and Y. Li "A fast

inter-prediction algorithm for HEVC based on temporal and spatial correlation,” *Multimedia Tools and Applications*, vol. 74, no. 24, pp. 11023-11043, Aug. 2014.

- [6] J. Nightingale, Qi Wang, C. Grecos, “Priority based methods for reducing the impact of packet loss on HEVC encoded video streams,” in *Proc. SPIE8656, Real-Time Image and Video Processing*, San Francisco, USA, Feb. 2013.
- [7] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, “An effective CU size decision method for HEVC encoders,” *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 465-470, Feb. 2013.
- [8] H. L. Tan, F. Liu, Y. H. Tan, and C. Yeo, “On fast coding tree block and mode decision for high-efficiency video coding (HEVC),” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 825-828, Kyoto, Japan, Mar. 2012.
- [9] S. Ahn, B. Lee, and M. Kim, “A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 422-435, Mar. 2015.
- [10] R.-H. Gweon, Y.-L. Lee, and J. Lim, “Early Termination of CU Encoding to Reduce HEVC Complexity,” document Joint Collaborative Team on Video Coding, document ISO/IEC and ITU-T, JCTVC-F045, Turin, Italy, Jul. 2011.
- [11] K. Choi, S.-H. Park, and E. S. Jang, “Coding Tree Pruning Based CU Early Termination,” document Joint Collaborative Team on Video Coding, document ISO/IEC and ITU-T, JCTVC-F092, Turin, Italy, Jul. 2011.
- [12] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, “Early SKIP Detection for HEVC,” document Joint Collaborative Team on Video Coding, document ISO/IEC and ITU-T, JCTVC-G543, Geneva, Switzerland, Nov. 2011.
- [13] HM Reference Software 14.0 [online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware

— 저 자 소 개 —



황인서(정회원)
2014년 아주대학교 전자공학 학사 졸업
2016년 아주대학교 전자공학 석사 졸업
2016년~현재 LIG 넥스원 재직

<주관심분야: 멀티미디어 코덱, 멀티미디어 신호처리, 멀티미디어용 ASIP설계, 컴퓨터 비전>



선우명훈(평생회원)
1980년 서강대학교 전자공학 학사 졸업.
1982년 한국과학기술원 전자공학 석사 졸업.
1982년~1985년 한국전자통신연구소(ETRI) 연구원

1985년~1990년 Univ. of Texas at Austin 전자공학 박사.
1990년~1992년 Motorola, DSP Chip Division (미국)
1992년~1996년 아주대학교 전기전자공학부 조교수
1996년~2001년 아주대학교 전자공학부 부교수
2001년~현재 아주대학교 전자공학부 교수
2011년~현재 IEEE Fellow Member
2011년~현재 IEEE CASS Board of Governor
2012년~2014년 대한전자공학회 반도체 소사이어티 회장
2014년~현재 대한전자공학회 반도체 소사이어티 자문위원
2016년~현재 아주대학교 ITRC uDSD 연구센터장
<주관심분야: ICT SoC 설계, ASIP 설계, 저전력 의료기기 디바이스 설계>