

논문 2014-51-3-14

# 쿼드트리 구조와 SATD를 이용한 HEVC 인코더의 고속 인트라 예측 방식

( A Fast Intra Prediction Method Using Quadtree Structure and SATD  
in HEVC Encoder )

김 영 조\*, 김 재 석\*\*

( Youngjo Kim and Jaeseok Kim<sup>©</sup> )

## 요 약

본 논문은 HEVC(high efficiency video coding) 인코더의 인코딩 시간을 줄이기 위한 고속 인트라 예측 방식을 제안한다. 제안하는 고속 인트라 예측 방식은 쿼드트리 구조와 SATD(Sum of Absolute Transformed Differences)를 사용한다. HEVC는 8×8 이상의 블록에서 SATD 값을 구하기 위해 8×8 hadamard 변환을 이용한 8×8 SATD 값을 사용한다. 제안하는 방식은 16×16 이상의 블록에서 각각의 8×8 SATD 결과를 이용해서 최적 SATD 값을 산출한다. 그 후, RDO를 위한 후보 모드의 SATD와 산출된 최적 SATD의 비교를 기반으로 후보 모드를 제거한다. 후보 모드를 제거함으로써 제안하는 방식은 RDO의 연산을 줄이고 전체 인코딩 시간을 줄이게 된다. 제안하는 방식은 8×8 블록에서는 추가로 4×4 SATD를 사용하여 최적 SATD를 구한다. 실험 결과 제안하는 방식은 거의 압축 성능 손실 없이 HM 12.1에 비해 5.33%의 인코딩 시간 감소 효과를 얻을 수 있었다.

## Abstract

This paper proposes a fast intra prediction method to reduce encoding time for the HEVC(high-efficiency video coding) encoder. The proposed fast Intra prediction method uses quadtree structure and SATD(Sum of Absolute Transformed Differences). In HEVC, a 8×8 SATD value using 8×8 hadamard transform is used to calculate a SATD value for 8×8 or larger blocks. The proposed method calculates the best SATD value by using each 8×8 SATD result in 16×16 or larger blocks. After that, the proposed method removes a candidate mode for RDO(Rate-Distortion Optimization) based on comparing SATD of the candidate mode and the best SATD. By removing candidate modes, the proposed method reduces the operation of RDO and reduces total encoding time. In 8×8 block, the proposed method uses additional 4×4 SATD to calculate the best SATD. The experimental results show that the proposed method achieved 5.08% reduction in encoding time compared to the HEVC test model 12.1 encoder with almost no loss in compression performance.

**Keywords :** HEVC, Fast intra prediction, Video coding, Video encoder

\* 학생회원, \*\* 정회원, 연세대학교 전기전자공학과  
(Department of Electrical and Electronic  
Engineering, Yonsei University)

© Corresponding Author(E-mail: jaekim@yonsei.ac.kr)

※ 본 연구는 산업통상자원부 및 한국산업기술평가관  
리원의 산업원천기술개발사업(정보통신)의 일환으로  
수행하였음. [10035389, 고해상도 영상정보 수집용  
Gbps급 초고속 초저전력 무선통신 SoC 연구]  
접수일자: 2013년12월24일, 수정완료일: 2014년2월26일

## I. 서 론

디지털 컴퓨터, 반도체, 신호처리 기술 등이 발전하면  
서, 디지털 방송이나 멀티미디어 통신 등의 멀티미디어  
서비스 역시 발전하고 있다. 그 중에서도 시각과 청각  
자료 등의 많은 정보를 가지는 비디오 신호는 중요한  
멀티미디어 중 하나이다. 이러한 비디오 신호는 많은

정보를 가지고 있는 만큼 매우 큰 용량을 가지고 있다. 따라서 비디오 신호를 포함하는 멀티미디어 데이터를 실생활에 적용하기 위해서는 비디오 데이터를 효과적으로 압축할 수 있는 방법이 필요하다. 이 비디오 압축 방식을 비디오 코덱이라고 부르며, ISO-IEC/MPEG과 ITU-T VCEG의 JVT(Joint Video Team)가 제안한 H.264/AVC<sup>[1]</sup> 표준이 현재 주로 사용되는 가장 좋은 성능을 가지는 비디오 코덱 표준이다. 그러나 최근 고화질, 고해상도 영상에 대한 요구가 증가함에 따라 큰 용량을 가지는 비디오 데이터의 사용이 증가하면서 기존의 비디오 코덱 표준(H.264/AVC)보다 높은 압축 효율을 갖는 새로운 비디오 압축 기술이 필요하게 되었다. 이런 흐름에 따라서 ISO-IEC/MPEG와 ITU-T/VCEG는 2010년에 공동 표준화를 진행하기 위한 JCT-VC (Joint Collaborative Team on Video Coding)를 구성하였다. JCT-VC는 H.264/AVC를 기반으로 압축효율의 2배 향상을 목표로 표준화 작업을 진행하여 2013년 3월 차세대 영상 부호화 표준인 HEVC(high efficiency video coding)<sup>[2]</sup>의 버전 1 최종표준안(FDIS: Final Draft International Standard)<sup>[3]</sup>을 배포하였다.

HEVC의 인트라 예측 부분에서는 압축 효율을 높이기 위한 많은 기술들이 새로 적용되었다.<sup>[4]</sup> 먼저 압축 성능을 높이기 위해서 planar 모드나 angular 모드 등이 추가 되고 전체 모드의 수가 증가하였다. 다음으로 대표적인 기술 중 하나는 쿼드트리 구조이다.<sup>[5]</sup> 기존 H.264/AVC의 인트라 예측 부분에서는 4×4부터 16×16의 세 개의 블록 크기 단위로 부호화 된 반면, HEVC에서는 최대 64×64에서 8×8까지 4개의 부호화 단위와 8×8 부호화 단위 안에서 4×4의 예측 단위까지 총 5개의 단위로 부호화 된다. 이와 같은 인트라 예측 모드의 증가와 쿼드트리 구조를 통해서 HEVC는 인트라 예측의 정확성을 크게 향상시키고 압축 효율을 높였다. 하지만 이러한 새로운 기술들의 추가로 많은 연산량이 추가되어 동영상 압축에 필요한 연산 시간을 증가시켜 실시간 인코딩을 하는데 큰 어려움을 준다. HEVC의 레퍼런스 소프트웨어인 HEVC Test Model (HM)<sup>[6]</sup>에서는 인트라 예측에서의 연산량 감소를 위하여 Hadamard 변환을 이용한 SATD(Sum of Absolute Transformed Differences)를 사용하여 고속 인트라 예측을 한다.<sup>[7]</sup> 하지만 아직 HEVC 인코더의 연산 시간의 문제를 해결하기에는 부족하다. 쿼드트리 구조로 인한

연산량을 감소시키는 고속화 방법도 제안 되었으나 상대적으로 높은 압축률 손실의 문제점을 가지고 있다.<sup>[8]</sup>

이 연산 시간의 문제를 해결하기 위해서 본 논문은 쿼드트리 구조와 SATD를 이용한 새로운 고속 인트라 예측 방식을 제안한다. 제안하는 방식은 HEVC의 기존 SATD 결과를 다시 사용하는 16×16 이상 블록에 대한 방식과 4×4 SATD 연산을 추가하는 8×8 블록에 대한 방식으로 나누어진다. 16×16 이상이 블록에서는 특별한 추가 연산 없이 적용되며 8×8 블록에서도 약간의 연산 추가로 적용이 가능하고 HM의 고속 인트라 예측 방식과 함께 적용되어 연산 시간을 효율적으로 줄이는 방식이다. HM의 고속 인트라 예측 방식은 SATD를 이용해서 전체 인트라 예측 모드 중에 일부 후보 모드를 선택해서 후보 모드에 대해서만 최종 인트라 예측을 함으로써 연산 시간을 감소시킨다. 제안된 방식은 HM의 고속 인트라 예측 방식을 통해서 선택된 후보 모드들 중에서 일부 모드를 추가로 제거해서 최종 후보 모드의 수를 더욱 줄이는 방식으로 연산 시간을 감소시킨다.

본 논문의 구성은 다음과 같다. II장에서는 기존 고속 인트라 예측 방식을 소개하고 제안 방식과의 알고리즘 접근의 차별성을 나타내며, III장에서는 기존 HEVC의 인트라 예측 방법에 대해서 살펴보고, IV장에서는 제안하는 방식에 대해서 소개한다. V장에서는 실험 결과를 보여주고, 마지막으로 VI장에서 결론을 맺는다.

## II. 기존 고속 인트라 예측 방식

HEVC 인트라 예측의 연산 시간의 문제를 해결하기 위해서 고속 인트라 예측 방식에 대한 많은 연구가 이루어지고 있다. Silva는 에지를 검출을 기반으로 하는 고속 인트라 모드 결정 방식을 제안하였다.<sup>[9]</sup> 이 방식은 4×4 블록 단위로 에지를 구하여 그 에지에 따라서 일부의 모드만 선택하여 인트라 예측을 진행하는 방식으로 고속 인트라 예측 방식을 구현하였다. 하지만 이 방식은 제안하는 방식에 비해 에지 검출을 위한 추가적인 알고리즘이 필요하며 속도 향상에 따른 압축률 손실이 비교적 높게 나타나는 문제가 있다. Zhang이 제안한 고속 인트라 예측 방식<sup>[10]</sup>은 제안하는 방식과 같이 SATD를 이용한 고속 인트라 예측 방식의 결과를 이용하지만 SATD 값을 직접 이용하지 않고 후보 모드의 선택 결과를 만들 사용하는 차이가 있다. 이 방식 역시 제안하

는 방식에 비해 높은 압축 손실을 나타낸다. Kim이 제안한 방식은 계층 구조를 이용한 고속 인트라 예측 방식이다.<sup>[11]</sup> 이 방식은 HEVC의 계층 구조를 이용하여 현재 계층과 상위 계층 사이의 관계와 예측 단위와 변환 단위 사이의 관계를 이용하여 하위 계층의 연산을 건너뛰는 방식과 상위 계층의 인트라 예측 모드들 이용하여 SATD를 이용한 고속 인트라 예측 방식을 통해 선정하는 후보 모드의 수를 줄이는 방식으로 고속 인트라 예측 방식을 진행한다. 이 방식은 본 논문에서 제안하는 방식과 비슷하게 계층적 구조에 대한 접근을 하였지만 이 방식은 계층적 구조에서 상위 계층의 결과를 이용하고, 본 논문에서 제안하는 방식은 하위 계층의 결과를 예측 한다는 것에서 차이가 있으며, 이 방식 역시 최종 결과에서 제안하는 방식에 비해 높은 압축 손실을 나타낸다.

### III. HEVC의 인트라 예측 방법

HM에서 부호화 단위나 예측 단위의 인트라 예측 방식은 크게 두 부분으로 나눌 수 있다. 고속 인트라 예측 방식인 RMD(Rough Mode Decision) 부분과 최종 인트라 예측 모드를 결정하는 RDO(Rate-Distortion Optimization) 부분이다.

먼저 RMD에서는 전체 35개의 인트라 예측 모드에 대해서 RDO에서 사용되는 SSE(Sum of Squared Error)를 사용하는 대신 그보다 좀 더 간단한 SATD를 이용하여 RDO에서 최종 인트라 예측 모드 결정을 위한 개략적인 후보 모드 군을 생성한다. RDO에서의 SSE의 연산을 줄여서 연산 시간을 단축 할 수 있게 하는 것이다. SATD를 구하기 위한 Hadamard 변환은 4x4 변환과 8x8 변환의 두 가지 크기의 변환이 이용된다. 4x4 블록에서는 4x4 변환이 사용되고 나머지 8x8 이상의 블록에서는 8x8 변환이 사용된다. RMD를 통해서 상대적으로 복잡한 영역에서 많이 선택되는 8x8 이하의 블록에서는 8개의 후보 모드가 선택되고, 16x16 이상의 블록에서는 3개의 후보 모드를 선택하게 된다. 이때, 주변 블록의 인트라 예측 모드로부터 얻은 최고 확률 모드가 후보 모드에 포함되지 않은 경우에는 최고 확률 모드를 후보 모드에 추가해 준다.<sup>[12]</sup> 그 후 RDO에서는 SSE를 이용한 Rate-distortion 기반 모드 결정을 수행하여 최종 인트라 예측 모드를 결정하게 된다.

위와 같은 방법으로 각 블록 별로 최종 인트라 예측 모드와 최종 코스트를 구한 후 그림 1과 같이 쿼드트리 구조로 각 블록의 코스트를 비교하여 가장 큰 부호화 단위인 64x64 블록 단위로 인트라 예측을 진행한다. 먼저 4x4 블록 4개로 부호화할 경우의 코스트와 8x8 블록 하나로 부호화할 경우의 코스트를 비교하여 8x8 블록의 최적 부호화 방법과 코스트를 구한다. 다음으로 하나의 8x8 블록 또는 4개의 4x4 블록 중 최적 부호화 방법으로 부호화 된 4개의 8x8 블록들과 16x16블록 하나로 부호화 할 경우의 코스트를 비교하여 16x16블록의 최적 부호화 방법과 코스트를 구한다. 마찬가지로 방법으로 32x32 블록과 비교하고 마지막으로 64x64 블록

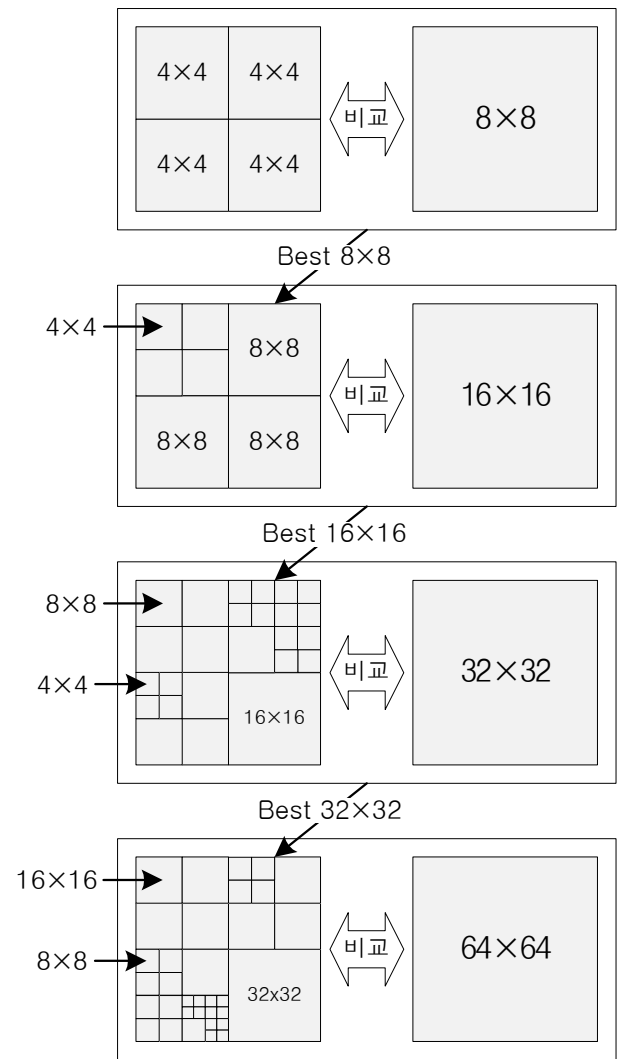


그림 1. HEVC에서 인트라 예측 방법의 쿼드트리 구조  
Fig. 1. The quadtree structure of intra prediction in HEVC.

과 비교하여 최종적으로 가장 큰 부호화 단위의 하위 블록 크기 구조 및 각 하위 블록의 최종 예측 모드를 결정 한다.

#### IV. 제안하는 방식

본 논문에서는 고속 인트라 예측 방식을 위하여, 블록 크기에 따라 두 가지 방법을 적용하여 예측 속도를 향상 시키는 방식을 제안한다. 4개 이상의 8x8 변환을 이용하여 SATD를 구하는 16x16 이상의 블록에서는 8x8 변환의 결과를 그대로 이용하는 방식을 사용하며, 하나의 8x8 변환을 이용하는 8x8 블록에서는 4개의 4x4 변환을 추가로 하는 방식을 사용한다.

32x32 Block

8x8 SATD 1	8x8 SATD 2	8x8 SATD 3	8x8 SATD 4
8x8 SATD 5	8x8 SATD 6	8x8 SATD 7	8x8 SATD 8
8x8 SATD 9	8x8 SATD 10	8x8 SATD 11	8x8 SATD 12
8x8 SATD 13	8x8 SATD 14	8x8 SATD 15	8x8 SATD 16

$$32 \times 32 \text{ SATD} = 8 \times 8 \text{ SATD } 1 + 8 \times 8 \text{ SATD } 2 + \dots + 8 \times 8 \text{ SATD } 16$$

그림 2. HM의 고속 인트라 예측 방식에서 32x32 블록의 SATD

Fig. 2. SATD of 32x32 block of fast intra prediction in HM.

##### 1. 16x16 이상 블록을 위한 고속 인트라 예측 방식

HM의 기존 고속 인트라 예측 방식에서 8x8 이상의 블록에서는 8x8 Hadamard 변환을 이용하여 SATD를 구한다. 즉, 16x16 블록에서는 4개의 8x8 변환이 사용되고 32x32 블록에서는 16개, 64x64 블록에서는 64개의 8x8 변환이 사용된다. 최종 SATD 결과는 각 8x8 단위의 SATD 결과를 독립적으로 구한 후 그 합으로 나타낸다. 그림 2는 HM의 기존 고속 인트라 예측 방식에서 32x32 블록의 SATD를 구하는 방식을 나타낸다.

본 논문에서 제안하는 방식은 위와 같이 16x16 이상의 블록에서 4개 이상의 8x8 SATD 결과가 사용되는 점을 이용한다. 제안하는 방식은 8x8 단위의 SATD 값의 합이 아닌 각각의 8x8 SATD 결과를 따로 뽑아서 이용한다. 8x8 SATD 단위로 35개의 모드에 대한 값을 비교하여 각 8x8 SATD 별로 최적의 코스트를 구하는 것이다. 구하는 식은 다음과 같다.

$$bS_8N = \min(S_8N(1), S_8N(2), \dots, S_8N(35)) \quad (1)$$

이때,  $bS_8N$ 는 N번째 8x8 SATD의 최적 코스트를 나타내며  $S_8N(m)$ 은 인트라 모드가 m 일 때의 N번째 8x8 SATD 코스트를 나타낸다. 그림 2를 통해서 32x32 블록에서의 예를 들면,  $S_8A(10)$ 은 인트라 모드가 10일 때 4번째 블록인 8x8 SATD 4의 값을 나타내는 것이다. 이와 같은 방법으로 16x16 블록에서는 4개 32x23 블록에서는 16개 64x64블록에서는 64개의 8x8 단위의 SATD 최적 코스트들을 구한다. 각 8x8 SATD의 최적 값을 모두 더하여 최종적으로 최적 SATD 값을 다음과 같이 구한다.

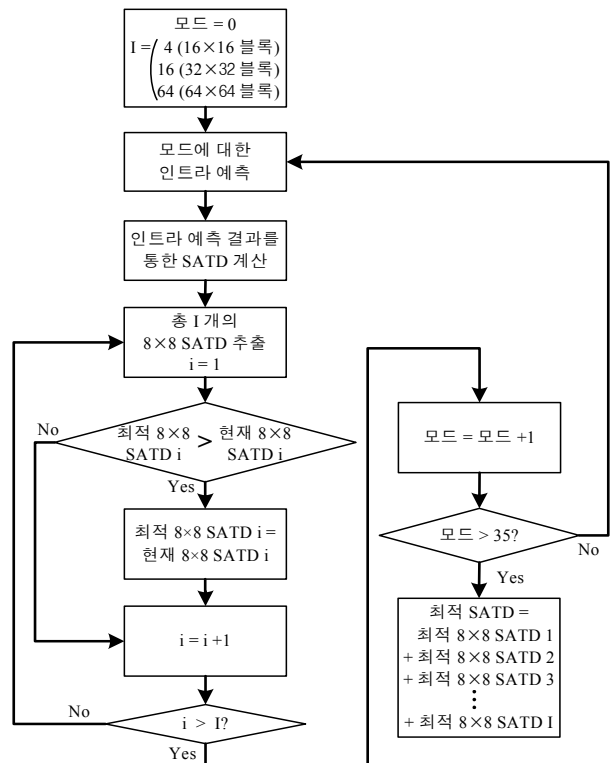


그림 3. 제안하는 방식의 최적 SATD 산출 과정  
Fig. 3. The calculating process of the best SATD in the proposed method.

$$bestSATD = \sum_{i=1}^N bS_i \quad (2)$$

이때  $N$ 은  $16 \times 16$  블록에서는 4,  $32 \times 32$  블록에서는 16,  $64 \times 64$  블록에서는 64이다. 그림 3은 제안하는 방식에서 최적 SATD 결과를 산출하는 과정을 나타낸다.

최적 SATD 값을 구한 후 제안하는 방식은 HEVC의 쿼드트리 구조의 성질을 이용한다. 그림 1에서 설명 했듯이 HEVC의 인트라 예측에서는 각 블록 크기 별로 인트라 예측을 진행하여 예측 결과와 코스트를 계산한 후에 각 블록 크기 별로 최종 코스트를 비교하여 작은 코스트를 갖는 블록 크기로 결정이 되는 구조를 가지고 있다. 이와 같은 쿼드트리 구조에서는 결국 최종 블록 크기로 선택되지 않은 블록들의 결과는 필요 없는 값이 된다. 즉, 이런 블록에 대해서는 인트라 예측을 하지 않아도 성능 손실은 발생하지 않는다.

앞 단락에서 설명한 방식으로 각  $8 \times 8$  SATD 단위로 최적 SATD 값을 구하면, 이 결과를 통해  $16 \times 16$  이상의 블록에서  $8 \times 8$  블록의 인트라 예측 결과를 대략적으로 추정해 볼 수 있다. 즉, 실제 인트라 예측을 진행하기 이전에 대략적으로 현재 블록의 인트라 예측 결과와  $8 \times 8$  블록의 인트라 예측 결과를 비교해 볼 수 있게 된다.

$16 \times 16$  이상의 블록에서는 RMD와 최고 후보 모드 MPM(most probable mode)를 용하여 3개에서 5개의

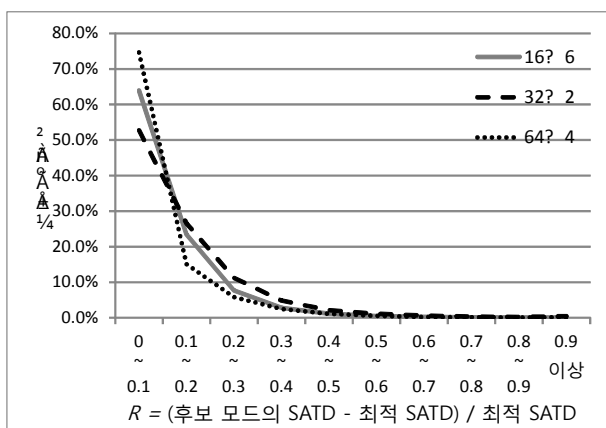


그림 4. 최적 SATD에 대한 후보 모드의 SATD 비율에 따른 후보 모드가 최종 모드로 선택되는 비율  
Fig. 4. The occurrence ratio of the candidate mode selected to final intra mode according to the ratio of SATD of a candidate mode to best SATD.

후보 모드를 선택하게 된다. 제안하는 방식은 이 후보 모드들의 SATD 결과 값과 제안하는 방식에서 구한 최적 SATD 값을 비교한다. 이 때, 최적 SATD 값이 어떤 후보 모드의 SATD 값에 비해서 많이 작은 경우에는 이 후보 모드가 현재 블록 사이즈에서 최종 인트라 예측 모드로 선택되어도  $8 \times 8$  블록의 예측 결과보다 큰 코스트를 가져 최종 블록으로 선택되지 않을 가능성이 매우 크게 된다. 즉, 이 후보 모드에 대한 인트라 예측 연산은 필요 없는 연산이 되는 것이다. 따라서 제안된 방식은 후보 모드의 SATD 값이 최적 SATD 값에 비해서 일정 비율 이상 큰 경우에는 후보 모드에서 제거한다. 결국 최적의 SATD 값을 구하기 위해 약간의 연산량이 증가 하였으나 최종적으로 불필요한 RDO 연산을 줄임으로써 전체적인 연산 속도를 향상 시킨다.

후보 모드의 SATD 값과 최적 SATD 사이의 비율 ( $R$ )은 다음과 같이 정의하여 사용하였다.

$$R = \frac{(\text{후보 모드의 SATD} - \text{최적 SATD})}{\text{최적 SATD}} \quad (3)$$

그림 4는  $16 \times 16$  이상의 각 블록 크기에서 최적 SATD에 대한 후보 모드의 SATD의 비율  $R$ 에 따른 분석 그래프이다. 0.1 단위의 각 비율에서 그 비율을 갖는 후보 모드가 전체 인트라 예측의 최종 예측 결과로 선택되는 비율을 나타낸다. 그래프 결과는 HEVC 레퍼런스 소프트웨어 HM 12.1<sup>[6]</sup> 버전에서 JCT-VC에서 실험을 위해 작성한 공동 실험조건 문서<sup>[13]</sup>에 나와 있는 “All Intra - Main” 환경에서 시뮬레이션 한 결과이다. 22, 27, 32, 37 네 개의 QP(Quantization Parameter) 값과 공동 실험조건 문서<sup>[13]</sup>에 나와 있는 C, D, E 세 클래스의 11개의 테스트 영상이 사용되었다. 그래프를 보면 후보 모드의 SATD에 대한 최적 SATD 비율이 0.4 이상일 경우에는 전체 인트라 예측의 최종 결과로 거의

표 1. 제안하는 방식에서 제거될 수 있는 후보 모드의 비율

Table 1. The ratio of candidate modes that can be removed in proposed method.

블록 크기	제거 가능한 후보 모드의 비율 (%)
$16 \times 16$	17.5
$32 \times 32$	21.6
$64 \times 64$	38.1
평균	25.7

선택되지 않는다는 것을 확인 할 수 있다. 따라서 제안하는 방식은 후보 모드의 SATD에 대한 최적 SATD 비율이 0.4 이상일 경우에 그 모드는 후보 모드에서 제거한다. 표 1은 방금 위에서 말한 방식으로 제거될 수 있는 후보 모드의 비율을 나타낸다. 그림 4와 같은 실험 환경을 사용하였다. 표 1을 보면 제안하는 방식이 16×16 이상의 블록에서 평균 25.7%의 후보 모드를 제거함으로써 그 만큼의 RDO 연산량을 줄이고 연산 속도를 향상 시킬 수 있다는 것을 확인 할 수 있다. 그림 5는 제안하는 16×16 이상 블록을 위한 고속 인트라 예측 방식의 전체 과정을 나타낸다.

2. 8×8 블록을 위한 고속 인트라 예측 방식

본 논문에서는 8×8 블록을 위한 인트라 예측 방식을 함께 제안한다. HM의 RMD에서 8×8 블록은 SATD 값을 구할 때, 8×8 Hadamard 변환을 사용하기 때문에 4장 1절에서 말한 방식을 그대로 사용할 수 없다. 따라서 제안하는 방식은 8×8 블록의 RMD를 진행 할 때, 기존 8×8 변환과 함께 4개의 4×4 변환을 함께 사용한다. RMD를 통한 8개의 후보 모드를 선정 할 때에는 8×8 변환의 결과를 그대로 사용하고, 제안하는 방식에서 사

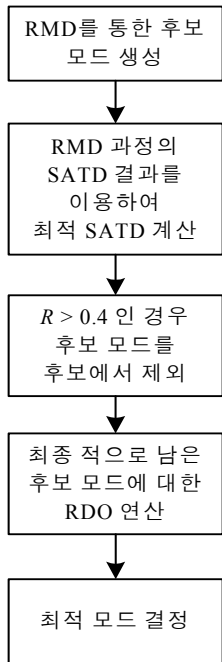


그림 5. 제안된 16×16 이상 블록을 위한 고속 인트라 예측 방식의 전체 과정

Fig. 5. Overall process of proposed fast intra prediction method for 16×16 Block.

용되는 최적 SATD 값을 구하기 위해서는 4개의 4×4 변환을 추가로 사용한다. 16×16 이상의 블록에서 각 8×8 SATD 별로 가장 작은 코스트를 구해서 최적 SATD 값을 구한 것과 마찬가지로 8×8 블록에서는 각 4×4 SATD 별로 가장 작은 코스트를 구해서 최적 SATD 값을 구한다. 수식으로 나타내면 다음과 같다.

$$bS_4N = \min(S_4N(1), S_4N(2), \dots, S_4N(35)) \quad (4)$$

$$bestSATD = \sum_{i=1}^4 bS_4i \quad (5)$$

이때,  $bS_4N$ 은 N번째 4×4 SATD의 최적 코스트를 나타내며  $S_4N(m)$ 은 인트라 모드가 m 일 때의 N번째 4×4 SATD 코스트를 나타낸다.

8×8 블록에서도 후보 모드의 SATD 값이 최적 SATD 값보다 클 경우 4×4 블록이 최종 인트라 예측 블록으로 선택될 가능성이 높으므로 후보 모드에서 제거해 주는 방식을 사용하게 된다. 즉, 8×8 블록에서도 최적 SATD 값을 구하기 위해 추가로 4개의 4×4 변환이 사용된다는 점을 제외하고는 같은 방식이 사용된다.

그림 6은 8×8 블록에서 후보 모드의 SATD에 대한 최적 SATD 비율에 따른 분석 그래프이며 그림 4와 같은 실험 환경에서 도출된 결과이다. 그래프 분석 결과 후보 모드의 SATD에 대한 최적 SATD 비율이 큰 부

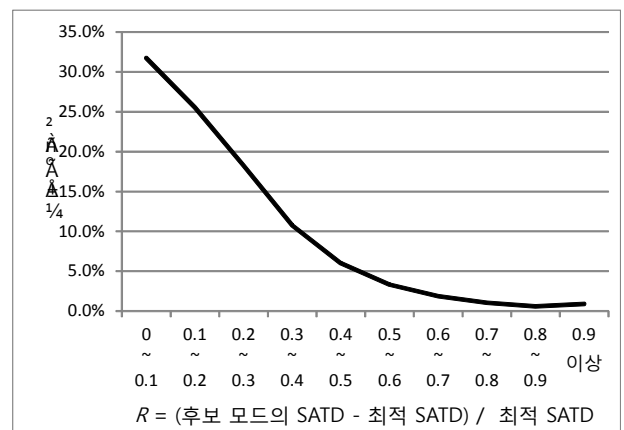


그림 6. 8×8 블록에서 후보 모드의 SATD에 대한 최적 SATD 비율에 따른 후보 모드가 최종 모드로 선택되는 비율

Fig. 6. The occurrence ratio of the candidate mode selected to final intra mode according to the ratio of best SATD to SATD of a candidate mode in 8×8 block.

표 2. 8×8 블록에 대해 제안하는 방식에서 제거될 수 있는 후보 모드의 비율

Table 2. The ratio of candidate modes that can be removed in proposed method for 8×8 block.

블록 크기	제거 가능한 후보 모드의 비율 (%)
8×8	20.4

분의 발생 빈도가 16×16 이상의 블록에 비해서 높게 나타나는 것을 볼 수 있다. 따라서 제안하는 방식은 8×8 블록에서는 후보 모드 제거를 위한 기준 값을 0.4 보다 높은 0.6으로 설정하여 0.6보다 높은 경우에 모드를 후보 모드에서 제거하게 된다. 표 2은 8×8 블록을 위한 고속 인트라 예측 방식으로 제거될 수 있는 후보 모드의 비율을 보여준다. 역시 그림 4와 같은 실험 환경을 사용하였다. 제안하는 방식이 8×8 블록에서 20.4%의 RDO 연산량을 줄이고 연산 속도를 향상 시킬 수 있음을 확인할 수 있다. 전체 프로세스는 그림 5와 같으며 R에 대한 기준 값만 0.4에서 0.6으로 바뀐다.

### V. 실험 결과

제안하는 고속 인트라 예측 방식의 검증 을 위해서 우리는 HEVC 레퍼런스 소프트웨어인 HM 12.1<sup>[6]</sup>을 사용하였다. 실험 환경 구성은 공동 실험 조건 문서<sup>[13]</sup>를 참조하였으며 자세한 내용은 표 3에 나타내었다. 제안하는 방식의 성능 검증을 위하여 우리는 제안하는 방식의 전체 인코딩 시간과 HM 12.1의 전체 인코딩 시간을 이용하였다. 두 인코딩 시간을 비교하여 제안하는 방식이 HM에 비해서 전체 인코딩 시간을 감소시키는 비율을 통해 성능을 나타내었다. 인코딩 시간 감소 비율( $\Delta$  Time)은 식 (1)과 같은 방식으로 계산하였다.

$$\Delta Time(\%) = \frac{(T_{pro} - T_{ref})}{T_{ref}} \quad (6)$$

식 (1)에서  $T_{pro}$ 는 제안하는 방식의 인코딩 시간이며,  $T_{ref}$ 는 HM 12.1의 인코딩 시간이다. 다음으로 압축 성능을 나타내기 위해서는 Bjontegaard-Delta(BD)<sup>[14]</sup> 측정법을 사용하였다. -x%의 BD-rate 값은 제안하는 방식이 기준 방식에 비해 x%의 전체 bit를 줄일 수 있다는 것을 의미하고 반대로 양수 값을 갖는 경우에는 전체 bit를 증가시켜 압축 효율이 떨어졌다는 것을 의미한다. HM 12.1과 비교한 제안하는 방식의 최종 성능은

표 3. 실험 조건 및 HM 구성

Table 3. Test conditions and HM configurations.

테스트 영상	<ul style="list-style-type: none"> <li>• Class A (2560×1600) : Traffic and People On Street</li> <li>• Class B (1920×1080) : Park Scene, Cactus</li> <li>• Class C (832×480) : Race Horses, BQ Mall, Party Scene, and Basketball Drill</li> <li>• Class D (416×240) : Race Horses, BQ Square, Blowing Bubbles, and Basketball Pass</li> <li>• Class E(1280×720) : FourPeople, Johnny, and KristenAndSara</li> </ul>
영상의 실험 프레임 수	10초의 비디오 재생을 위한 프레임 수
소프트웨어	HM 12.1
Quantization Parameter	22, 27, 32, 37
기본 구성 조건	All Intra - Main profile
컴퓨터 환경	CPU - Intel(R) Core(TM) i5 RAM - 8.00GB Parallel processing - 사용하지 않음

표 4와 표 5에 나타내었다. 표 4는 16×16이상 블록에 대한 고속 인트라 예측 방식만 적용된 결과이며, 표 5는 16×16이상 블록에 대한 고속 인트라 예측 방식과 8×8 블록에 대한 고속 인트라 예측 방식을 함께 적용한 경우의 성능을 나타낸다. 표 4의 실험 결과 16×16 이상의 블록에 대한 고속 인트라 예측 방식을 적용할 경우 전체 인코딩 시간이 평균 3.58% 줄어들며 속도 향상에 따른 압축 성능 손실은 평균 0.04%로 거의 손실이 발생하지 않는 것을 확인할 수 있다.

각 클래스 별로 비교하면 큰 영상 크기를 가지는 클래스가 속도 향상이 상대적으로 많이 되는 것을 확인할 수 있다. 영상 크기가 가장 큰 클래스 A에서 평균 6.69%로 가장 높은 속도 향상이 이루어 졌다. 이는 영상 프레임이 정확히 64×64의 블록으로 나누어지지 않기 때문에 발생하는 현상이다. HM에서는 영상 프레임의 테두리 부분에서는 테두리에 맞도록 64×64블록이 아닌 작은 블록으로 인코딩을 한다. 상대적으로 작은 사이즈의 영상에서는 이렇게 작은 블록으로 인코딩을 하는 테두리 부분의 블록 비율이 높게 된다. 첫 번째 제안하는 방식이 기본적으로 크기가 큰 16×16 이상의 블록들에 대해서 RDO 후보 모드 제거하여 연산량 감소를 통해서 속도 향상을 시키기 때문에 작은 블록으로

표 4. 16×16이상 블록을 위한 고속 인트라 예측 방식의 인코딩 시간 감소 비율 및 압축 성능

Table 4. Encoding time reduction ratio and compression performance of the fast intra prediction method for 16×16 or larger blocks.

Class	영상	HM 12.1			제한된 방법 (8×8 미포함)			성능 비교				
		QP	Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	ΔTime (%)		
A	Traffic	22	101857.35	43.35	1.29	101881.35	43.35	1.22	0.04	-5.50		
		27	57380.19	40.16	1.11	57400.22	40.16	1.06		-4.64		
		32	32742.57	37.12	0.99	32743.65	37.12	0.94		-4.69		
		37	18509.52	34.10	0.90	18504.37	34.10	0.87		-3.38		
	PeopleOnStreet	22	104688.17	43.26	1.32	104780.73	43.26	1.19	0.13	-10.11		
		27	60852.47	39.82	1.13	60874.42	39.81	1.03		-9.03		
		32	34778.63	36.70	1.01	34800.91	36.69	0.92		-8.88		
		37	20386.04	33.83	0.92	20397.38	33.83	0.85		-7.26		
	B	ParkScene	22	52729.06	41.68	1.15	52732.57	41.68	1.14	0.01	-0.88	
			27	28587.68	38.60	0.94	28592.37	38.60	0.93		-0.22	
			32	14837.46	35.61	0.81	14838.30	35.61	0.81		0.18	
		37	7299.99	32.79	0.71	7298.59	32.79	0.71	0.66			
Cactus		22	105375.06	40.57	2.49	105405.82	40.57	2.36	0.03	-4.91		
		27	48706.70	37.93	1.95	48712.89	37.93	1.89		-3.05		
	32	26368.31	35.69	1.64	26366.05	35.69	1.60	-2.66				
C	BasketballDrill	22	20807.22	41.76	0.48	20812.04	41.76	0.45	0.01	-5.12		
		27	11314.06	38.37	0.39	11315.43	38.37	0.38		-3.52		
		32	6076.51	35.41	0.34	6077.11	35.41	0.32		-3.82		
	BQMall	27	23372.63	41.92	0.55	23374.01	41.92	0.52	0.02	-2.75		
		27	13971.20	39.02	0.46	13974.13	39.02	0.43		-6.19		
		32	8221.12	35.97	0.41	8223.16	35.96	0.39		-5.78		
	PartyScene	37	4694.38	32.89	0.37	4696.48	32.89	0.35	0.00	-5.51		
		22	43865.06	41.08	0.58	43867.93	41.08	0.57		-4.07		
		27	27228.14	36.87	0.50	27228.57	36.87	0.51		-1.60		
	RaceHorses	32	16226.49	33.06	0.43	16225.68	33.06	0.43	0.00	1.01		
		37	8798.40	29.37	0.36	8799.14	29.38	0.36		0.05		
		22	14997.78	42.29	0.28	14996.85	42.28	0.27		0.18		
	D	BasketballPass	27	9001.64	38.91	0.24	9001.69	38.91	0.23	0.02	-4.27	
			32	5108.75	35.49	0.21	5108.08	35.49	0.20		-3.53	
			37	2575.92	32.12	0.19	2576.69	32.12	0.18		-3.68	
		BQSquare	22	5321.22	42.95	0.11	5322.50	42.95	0.11	0.03	-3.45	
			27	3167.79	39.38	0.09	3168.87	39.38	0.09		-3.31	
			32	1816.54	35.98	0.08	1816.59	35.98	0.08		-3.01	
		BlowingBubbles	37	1012.37	32.82	0.07	1013.06	32.82	0.07	0.00	-2.43	
			22	13014.80	41.21	0.17	13015.09	41.21	0.16		-1.86	
			27	8254.75	36.81	0.14	8254.58	36.81	0.14		-2.32	
		E	Johnny	32	5076.88	33.00	0.12	5076.72	33.00	0.12	0.00	-1.76
				37	3013.52	29.45	0.11	3013.62	29.45	0.11		-1.69
				22	11454.17	41.03	0.14	11453.99	41.03	0.14		-1.99
RaceHorses	27		7070.80	36.72	0.12	7071.15	36.72	0.12	0.00	-1.20		
	32		4023.53	32.74	0.10	4022.80	32.74	0.10		-1.00		
	37		2072.65	29.21	0.09	2072.24	29.21	0.09		-0.27		
E	FourPeople	22	4459.80	42.42	0.07	4459.91	42.42	0.07	0.01	-0.42		
		27	2684.23	38.41	0.06	2684.59	38.41	0.06		-3.96		
		32	1475.36	34.58	0.05	1475.33	34.58	0.05		-3.77		
	Johny	37	743.95	31.28	0.05	744.53	31.29	0.05	0.12	-2.90		
		22	30226.20	43.80	1.05	30233.04	43.80	0.98		-2.92		
		27	18534.27	41.23	0.92	18538.18	41.23	0.86		-6.10		
		32	11401.86	38.36	0.84	11403.68	38.36	0.79		-6.41		
		37	6930.37	35.27	0.78	6930.17	35.26	0.75		-6.62		
		22	20139.58	44.05	0.96	20159.17	44.05	0.91		-4.07		
	KristenAndSara	27	11231.64	41.84	0.84	11241.92	41.84	0.80	0.08	-5.63		
		32	6424.09	39.50	0.78	6429.72	39.50	0.74		-4.83		
		37	3710.49	36.90	0.73	3711.82	36.89	0.71		-5.50		
22		22190.88	44.42	0.96	22198.66	44.42	0.90	-3.66				
27		13258.55	42.10	0.87	13263.90	42.10	0.81	-6.46				
32		8031.88	39.46	0.80	8034.91	39.46	0.76	-6.93				
평균									0.04	-3.58		

표 5. 8×8 블록을 위한 고속 인트라 예측 방식이 포함된 방식의 인코딩 시간 감소 비율 및 압축 성능

Table 5. Encoding time reduction ratio and compression performance of the method including the fast intra prediction method for 8×8 block.

Class	영상	HM 12.1			제한된 방법 (8×8 포함)			성능 비교				
		QP	Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	ΔTime (%)		
A	Traffic	22	101857.35	43.35	1.31	101888.10	43.34	1.21	0.29	-7.97		
		27	57380.19	40.16	1.11	57413.63	40.15	1.02		-7.88		
		32	32742.57	37.12	0.99	32779.40	37.11	0.92		-6.26		
		37	18509.52	34.10	0.90	18563.23	34.09	0.86		-5.15		
	PeopleOnStreet	22	104688.17	43.26	1.33	104981.19	43.24	1.14	0.78	-13.98		
		27	60852.47	39.82	1.13	61051.58	39.80	0.98		-13.20		
		32	34778.63	36.70	1.01	34947.01	36.67	0.91		-10.44		
		37	20386.04	33.83	0.91	20549.03	33.82	0.82		-10.17		
	B	ParkScene	22	52729.06	41.68	1.17	52730.12	41.68	1.12	0.13	-4.16	
			27	28587.68	38.60	0.94	28597.06	38.59	0.92		-1.29	
			32	14837.46	35.61	0.81	14846.16	35.61	0.79		-1.52	
		37	7299.99	32.79	0.72	7309.75	32.79	0.72	-0.48			
Cactus		22	105375.06	40.57	2.50	105404.68	40.57	2.37	0.09	-4.84		
		27	48706.70	37.93	1.95	48713.83	37.92	1.86		-4.55		
	32	26368.31	35.69	1.64	26370.86	35.69	1.60	-2.41				
C	BasketballDrill	37	14280.49	33.26	1.49	14280.34	33.26	1.49	0.10	-2.08		
		22	20807.22	41.76	0.48	20813.70	41.76	0.44		-8.60		
		27	11314.06	38.37	0.39	11319.28	38.37	0.37		-6.28		
	BQMall	32	6076.51	35.41	0.34	6083.71	35.41	0.32	0.12	-5.82		
		37	3356.01	32.77	0.30	3362.77	32.77	0.29		-3.83		
		22	23372.63	41.92	0.55	23375.35	41.91	0.50		-8.94		
	PartyScene	27	13971.20	39.02	0.46	13975.74	39.02	0.43	0.03	-6.96		
		32	8221.12	35.97	0.41	8225.76	35.96	0.38		-6.41		
		37	4694.38	32.89	0.37	4703.31	32.88	0.35		-4.67		
	RaceHorses	22	43865.06	41.08	0.58	43867.48	41.07	0.56	0.09	-2.90		
		27	27228.14	36.87	0.50	27226.88	36.87	0.48		-4.12		
		32	16226.49	33.06	0.43	16229.48	33.06	0.41		-3.12		
	D	BasketballPass	37	8798.40	29.37	0.37	8805.11	29.38	0.36	0.09	-1.56	
			22	14997.78	42.29	0.29	14997.01	42.28	0.27		-6.61	
			27	9001.64	38.91	0.24	9002.94	38.91	0.23		-6.47	
		BQSquare	32	5108.75	35.49	0.21	5109.96	35.49	0.20	0.15	-4.96	
			37	2575.92	32.12	0.18	2579.98	32.11	0.18		-1.07	
			22	5321.22	42.95	0.11	5323.80	42.94	0.10		-6.57	
		BlowingBubbles	27	3167.79	39.38	0.10	3169.32	39.38	0.09	0.02	-5.72	
			32	1816.54	35.98	0.08	1817.86	35.98	0.08		-3.74	
			37	1012.37	32.82	0.07	1015.06	32.81	0.07		-2.59	
		E	Johnny	22	13014.80	41.21	0.17	13015.33	41.21	0.16	0.03	-4.96
				27	8254.75	36.81	0.14	8254.99	36.81	0.14		-4.98
				32	5076.88	33.00	0.12	5078.54	33.00	0.12		-3.64
RaceHorses	37		3013.52	29.45	0.11	3015.62	29.45	0.10	0.18	-2.56		
	22		11454.17	41.03	0.15	11453.94	41.02	0.14		-3.54		
	27		7070.80	36.72	0.12	7071.36	36.72	0.12		-3.14		
E	FourPeople	32	4023.53	32.74	0.11	4022.91	32.74	0.10	0.29	-2.09		
		37	2072.65	29.21	0.09	2073.27	29.21	0.09		-0.33		
		22	4459.80	42.42	0.07	4460.82	42.41	0.07		-6.31		
	Johny	27	2684.23	38.41	0.06	2685.65	38.40	0.06	0.31	-5.56		
		32	1475.36	34.58	0.05	1476.84	34.57	0.05		-4.92		
		37	743.95	31.28	0.05	746.33	31.28	0.04		-4.02		
		22	30226.20	43.80	1.04	30228.71	43.79	0.94		-10.29		
		27	18534.27	41.23	0.92	18543.88	41.22	0.84		-8.46		
		32	11401.86	38.36	0.84	11419.22	38.35	0.78		-6.81		
	KristenAndSara	37	6930.37	35.27	0.79	6954.37	35.26	0.73	0.24	-6.58		
		22	20139.58	44.05	0.96	20160.77	44.04	0.90		-7.09		
		27	11231.64	41.84	0.84	11247.14	41.84	0.80		-5.39		
32		6424.09	39.50	0.77	6439.17	39.49	0.74	-4.22				
37		3710.49	36.90	0.74	3724.45	36.89	0.71	-4.08				
22		22190.88	44.42	0.96	22199.95	44.41	0.90	-6.44				
평균									0.19	-5.33		



인코딩을 할 경우 속도 향상을 이루기 어렵다. 따라서 프레임 사이즈가 큰 영상들이 테두리 부분의 작은 블록 인코딩 비율이 상대적으로 작기 때문에 제안된 방식에 의해서 보다 많은 속도 향상 효율을 나타낸다.

다음으로 QP에 따른 결과를 보면 QP가 낮을수록 높은 속도 향상을 보이는 것을 확인 할 수 있다. QP 22에서 평균 4.75%의 속도 향상을 이루었으며, QP 37에서는 평균 2.75%의 속도 향상을 이루었다. QP가 낮을수록 영상이 더욱 선명하고 복잡해지며 반대로 QP가 높을수록 영상의 화질이 흐려지고 뭉개지게 되는데 이 때문에 낮은 QP에서는 상대적으로 작은 블록으로 인코딩 되는 경우가 많아지게 된다. 따라서 QP가 낮으면 제안하는 방식에서 예측한 8×8 블록의 코스트가 16×16 이상 블록의 코스트보다 충분히 작은 경우가 많이 발생하여 QP가 큰 경우에 비해 더 많은 RDO 후보 모드를 제외하여 더 높은 연산 속도 향상을 이룰 수 있게 된다.

표 5는 8×8 블록을 위한 고속 인트라 예측 방식을 포함한 제안하는 방식 전체를 적용한 성능을 나타낸다. 실험 결과 제안하는 방식은 HM 12.1에 비해 전체 인코딩 시간이 평균 5.33%가 짧아졌으며, 속도 향상에 따른 압축 성능 손실은 평균 0.19%로 약간 증가하였지만 여전히 무시할 만큼 작은 성능 손실을 보인다. 8×8 블록을 위한 고속 인트라 예측이 포함되며 1.75%의 연산 시간이 추가로 감소하고 0.15%의 압축 성능 손실이 증가한 것을 확인 할 수 있다. 8×8 블록에 대한 고속 인트라 예측 방식은 첫 번째 제안된 방식인 16×16 이상 블록에 대한 고속 인트라 예측 방식에 비해서 속도 향상 및 압축 성능 손실 부분에서 상대적으로 안 좋은 결과를 보이고 있다. 이는 8×8에서 4×4 SATD 연산을 추가하여 연산량 감소 효과가 줄었으며, 4×4 SATD와 8×8 SATD의 서로 다른 SATD 결과를 비교하다 보니 상대적으로 낮은 상관관계를 가지게 되어 그림 6에서 나타난 것과 같이 0.6의 높은 기준 값에도 상대적으로 많은 최종 선택 모드들이 후보 모드에서 제외됨으로 인하여 압축 성능 손실이 증가하게 된 것이다. 하지만 압축 손실은 여전히 무시할 만큼 작아 인트라 예측의 속도 향상을 위한 고속 알고리즘으로 충분히 활용될 수 있을 것으로 보인다.

이 실험 결과에서도 역시 클래스 A에서 가장 큰 속도 향상을 보여 평균 9.38%의 연산 시간을 감소시켰으며 QP에 따라서는 QP 22에서 평균 7.11%로 가장 크게

표 6. 기존 고속 인트라 예측 방식과 제안하는 방식의 성능 비교

Table 6. The comparison between the performance of the existing fast intra prediction method and the performance of the proposed method.

	기존 방식			제안 방식
	Silva's	Zhang's	Kim's	
평균 인코딩 시간 감소(%)	18.88	14.80	22.99	5.33
평균 압축 성능 감소(%)	1.30	0.64	0.95	0.19
압축 성능 감소에 대한 인코딩 시간 감소 비율	14.52	23.13	24.2	28.05

연산 시간을 감소 시켰다.

표 6은 II장에서 소개한 기존의 고속 인트라 예측 방식과 제안 방식의 성능을 비교하여 나타낸 것이다. 제안 방식이 평균 압축 성능 감소 부분에서 매우 높은 성능을 보인다. 평균 인코딩 속도 부분에서는 상대적으로 제안 방식의 성능이 낮지만 압축 성능 감소에 대한 인코딩 시간 감소의 효율로 보았을 경우에는 가장 높은 효율을 나타내고 있다. 즉, 제안 방식은 기존 고속 인트라 예측 방식에 비해 더욱 효율적인 고속 인트라 예측 방식이라는 것을 확인 할 수 있다.

## VI. 결 론

본 논문에서는 쿼드트리 구조와 SATD를 이용한 HEVC 인코더의 고속 인트라 예측 방식을 제안하였다. 제안 방식은 16×16 이상의 블록에 대해서 사용되는 HEVC의 기존 고속 인트라 예측 방식에서 계산되는 SATD 값을 그대로 이용한 방식과 8×8 블록에서 사용되는 추가로 4×4 SATD 연산을 사용하는 두 가지 방식으로 이루어 졌다. 실험 결과 제안하는 방식은 압축 성능 손실이 거의 없이 평균 5%이상의 연산 시간을 감소 시켰다. 기존 고속 알고리즘과의 비교에서는 압축 성능 감소에 대한 연산 시간 감소의 비율이 가장 높게 나타났다. 이 실험 결과는 본 논문에서 제안하는 방식이 실시간 인코딩을 위한 고속 인트라 예측 방식의 좋은 대안이 될 수 있음을 보여준다.

이후 4×4를 포함한 모든 블록에 적용 가능한 고속 인트라 예측 방식에 대해서 연구할 계획이다.

## REFERENCES

- [1] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services", *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG4-AVC)*, 4th ed., Sept. 2008.
- [2] 호요성, 최정아 "UHD 고화질 영상 압축 기술", 진샘미디어, 1쪽-286쪽, 2013년
- [3] B. Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)", presented at JCTVC-L1003, *JCT-VC of ISO/IEC and ITU-T*, Geneva, CH, Jan. 2013.
- [4] Jani Lainema, Frank Bossen, Woo-Jin Han, Junghye Min, and Kemal Ugur, "Intra Coding of the HEVC Standard", *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 22, no. 12, Dec. 2012.
- [5] 서찬원, 한종기, "HEVC의 계층적 부호화 블록 구조", *전자공학회지*, 제38권 제8호, 22-26쪽, 2011년 8월.
- [6] HEVC Test Model 12 "HM 12.1", [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSofware/tags/HM-12.1](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSofware/tags/HM-12.1)
- [7] Y. Piao, J.H. Min, and J. Chen, "Encoder improvement of unified intra prediction", presented at JCTVC-C207, *JCT-VC of ISO/IEC and ITU-T*, Guangzhou, China, Oct. 2010.
- [8] SungEun Yoo, Yong-Jo Ahn, and DongGyu Sim, "Fast HEVC Encoding based on CU-Depth First Decision", *Journal of the Institute of Electronics Engineers of Korea - Signal Processing*, Vol.49-SP, No.3, page(s): 40-50, 2012.5
- [9] Thaisa L.da Silva, Luciano V. Agostini and Luis A. da Silva Curz, "Fast HEVC intra prediction mode decision based on EDGE direction information", *20th European Signal Processing Conference*, pp.1214-1218, Bucharest, Romania, Aug. 2012.
- [10] Mengmeng Zhang, Chuan Zhao and Jizheng Xu, "An adaptive fast intra mode decision in HEVC", *19th IEEE International Conference on Image Processing(ICIP)*, pp.221-224, Orlando, FL, USA, Sept. 2012.
- [11] Jaehwan Kim, Jungyoup Yang, Hoyoung Lee, and Byeungwoo Jeon, "Fast Intra Mode Decision of HEVC based on Hierarchical Structure," *Proc. of ICICS*, pp1-4, Singapore, Dec. 2011.
- [12] L. Zhao, L. Zhang, S. Ma, and D. Zhao, "Fast Mode Decision Algorithm for Intra Prediction in HEVC," *IEEE Visual Communications and Image Processing*, pp.1-4, Tainan, Taiwan, Nov. 2011.
- [13] F. Bossen, "Common test conditions and software reference configurations", presented at JCTVC-L1100, *JCT-VC of ISO/IEC and ITU-T*, Geneva, CH, Jan. 2013.
- [14] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-Curves", *ITU-T SG16 Q.6 VCEG*, Doc. VCEG-M33, 2001.

## 저 자 소 개



김 영 조(학생회원)  
2009년 연세대학교 전기전자  
공학과 학사 졸업.  
2009년~현재 연세대학교  
전기전자공학과  
석·박사 통합 과정.

<주관심분야 : 비디오 코딩, HEVC, SoC 설계>



김 재 석(정회원)  
1977년 연세대학교 전자공학과  
학사 졸업.  
1979년 KAIST 전기전자공학과  
석사 졸업.  
1988년 Rensselaer Polytechnic  
Institute, NY 전자공학과  
박사 졸업.

1988년~1993년 AT&T Bell Lab. Member of  
Technical Staff

1993년~1995년 한국전자통신연구원 책임연구원.  
1995년~현재 연세대학교 전기전자공학과 교수.

<주관심분야: 통신 및 영상 시스템, VLSI 신호  
처리, 임베디드 S/W 및 SoC 구현>