

논문 2014-51-8-12

# HEVC의 Transform Skip Mode를 위한 Rough Mode Decision 알고리즘

( A Rough Mode Decision Algorithm for Transform Skip Mode in  
HEVC )

김 영 조\*, 김 재 석\*\*

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

## 요 약

HEVC(high efficiency video coding) 표준에서 사용되는 기존의 rough mode decision(RMD) 알고리즘은 transform skip mode(TSM)와는 연관성이 낮은 DCT를 기반으로 하는 모드 선택 방법을 이용하고 있다. 따라서 기존 RMD의 결과로 구한 후보 모드가 TSM에서 사용될 때, 압축 효율이 손실되고 인코딩 시간을 낭비하게 된다. 본 논문은 HEVC에서의 TSM을 위한 새로운 RMD를 제안한다. 우리가 제안한 RMD 알고리즘은 TSM에서 최선의 모드를 선택할 확률을 높이는 새로운 비용 함수를 제안하여 코딩 효율을 향상시킨다. 또한, 제안하는 알고리즘은 새롭게 제안한 임계값을 기준으로 선택 가능성이 거의 없는 TSM의 인코딩 과정을 생략하여 인코딩 시간을 줄인다. 실험 결과 제안하는 방식은 HEVC 표준에 비해서 10%의 인코딩 시간을 줄이며 스크린 콘텐츠에 대해서 0.3%의 압축률을 향상시킨다.

## Abstract

The existing rough mode decision (RMD) algorithm, used in HEVC standard loses the coding efficiency and wastes encoding time when encoding using transform skip mode (TSM) because the RMD algorithm in HEVC uses a selection method that is designed for DCT, not for TSM. This paper proposes a new RMD algorithm for TSM in HEVC. Our proposed RMD algorithm enhances the coding efficiency by employing a new cost function to increase the probability of selecting the best intra prediction mode for TSM. In addition, it reduces the encoding time by skipping the encoding process of least feasible TSM based on a newly proposed threshold value. The experiment results show that the proposed method achieves coding gains of -0.3% for screen contents with a 10% reduction in encoding time compared to those of the HEVC standard.

**Keywords** : HEVC, Rough Mode Decision (RMD), Transform Skip Mode (TSM), intra prediction

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

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

※ 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신)의 일환으로 수행하였음. [10035389, 고해상도 영상정보 수집용 Gbps급 초고속 초저전력 무선통신 SoC 연구]  
접수일자: 2014년06월26일, 수정일자: 2014년07월16일  
수정완료: 2014년07월31일

## I. 서 론

HEVC(high efficiency video coding)<sup>[1]</sup>는 최근 증가한 HD-TV나 블루레이 등과 같은 고화질, 고해상도의 영상 콘텐츠를 제한된 대역폭에서 전송하거나 저장 등의 처리를 위해 개발된 차세대 비디오 코덱 표준이다. HEVC는 ISO-IEC/MPEG와 ITU-T/VCEG가 2010년에

공동 표준화를 진행하기 위해 구성된 JCT-VC(Joint Collaborative Team on Video Coding)에 의해서 개발되었으며, 기존 비디오 코덱 표준인 H.264/AVC<sup>[2]</sup>에 비해 압축효율의 2배 향상을 목표로 표준화 작업을 진행하여 2013년 3월 차세대 영상 부호화 표준인 HEVC의 버전 1 최종표준안(FDIS: Final Draft International Standard)<sup>[3]</sup>을 배포하였다.

최근 클라우드 서비스를 이용한 비즈니스 프로세스가 늘어감에 따라, 스크린 콘텐츠의 원격 디스플레이가 더욱 중요해졌다. 이러한 스크린 콘텐츠의 코딩 효율을 향상시키기 위해서 JCT-VC 그룹은 스크린 콘텐츠를 테스트 영상으로 추가 했으며, 코딩 효율을 높이기 위한 다양한 알고리즘을 제안했다. TSM(transform skip mode)<sup>[4]</sup>은 이렇게 스크린 콘텐츠를 위해 제안된 알고리즘 중 하나로 최종 HEVC 표준에 채택된 기술이다. TSM은 변환, 양자화, 역양자화, 역변환으로 이루어진 RDO(full rate-distortion optimization) 과정에서 DCT(discrete cosine transform)나 DST(discrete sine transform)의 변환과 역변환 과정을 생략하는 방식이다.

RMD(rough mode decision)는 HEVC에서 인트라 예측의 연산 속도 향상을 위한 고속화 방식이다.<sup>[5]</sup> 기존 H.264/AVC 표준에서는 9개의 인트라 예측 모드를 사용하는 반면에 HEVC에서는 35개의 인트라 예측모드를 사용하여 압축 효율을 높였으나 상대적으로 연산 시간이 많이 증가하게 되었다. 따라서 HEVC의 참조 소프트웨어 HEVC Test Model (HM)<sup>[6]</sup>에서는 인트라 예측에서의 연산량 감소를 위하여 RMD를 채택하여 사용하게 되었다. RMD는 RDO 과정 이전에 좀 더 단순한 함수를 사용하여 각 인트라 예측 모드의 비용을 예측하여 블록 크기에 따라 3개나 8개의 후보 모드를 선정된 후 선정된 모드들에 대해서만 RDO를 통한 최종 비용을 비교하여 최종 모드를 선택하는 방법으로 연산이 많은 RDO의 연산 시간을 줄이는 것으로 인트라 예측의 연산 시간을 줄이는 방식이다. 이때, 선택된 후보 모드들에 대해서 한번은 DCT나 DST의 변환을 이용한 방법으로 최종 비용을 구하고 다시 앞에서 도입된 TSM을 이용하여 최종 비용을 구하여 각 모드별로 두 개의 비용을 구하고 비교하는 방식이 사용된다. RMD는 예상 비용을 구하기 위해서 hadamard 변환을 이용한 SATD(Sum of Absolute Transformed Differences)를 사용한다.<sup>[7]</sup> Hadamard 변환은 간소화된 DCT와 비슷하며,

hadamard 변환을 이용하여 구한 비용은 DCT나 DST를 이용한 RD(rate-distortion) 비용과 높은 연관성을 갖는다. 하지만 hadamard 변환을 이용하여 구한 비용과 TSM을 이용한 RD 비용은 상대적으로 낮은 연관성을 갖는다. 결국 TSM을 이용한 인트라 예측에서는 RMD를 통해서 최적의 예측 모드를 선택할 확률이 매우 낮아지게 되고, 이것은 압축 효율을 떨어트림과 동시에 불필요한 예측 모드에 대한 인코딩으로 연산 시간 또한 증가하는 문제를 발생시킨다.

이와 같은 hadamard 변환을 이용한 RMD와 TSM 사이의 연관성 문제를 해결하기 위해 본 논문은 TSM을 위한 새로운 RMD 알고리즘을 제안한다. 제안하는 알고리즘은 TSM의 최적 예측모드를 선택할 확률을 증가시켜 압축 효율을 향상시키며, 임계값을 기반으로 TSM에서 최종적인 최적 예측모드가 될 가능성이 거의 없을 경우에는 TSM 부분을 인코딩을 하지 않고 건너뛰어서 연산 시간을 감소시킨다. 기존 고속 인트라 예측 알고리즘들은 대부분 연산 시간을 줄이는 대신 압축 효율이 감소하는 비용이 발생하지만, 제안하는 알고리즘은 일반 영상에서는 압축 효율 감소 없이 연산 시간만 감소하며, 스크린 콘텐츠에서는 연산 시간 감소와 더불어 압축 효율도 증가된다.

본 논문의 구성은 다음과 같다. II장에서는 기존 고속 인트라 예측 방법을 소개하고 제안 방법과 알고리즘 접근에서의 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

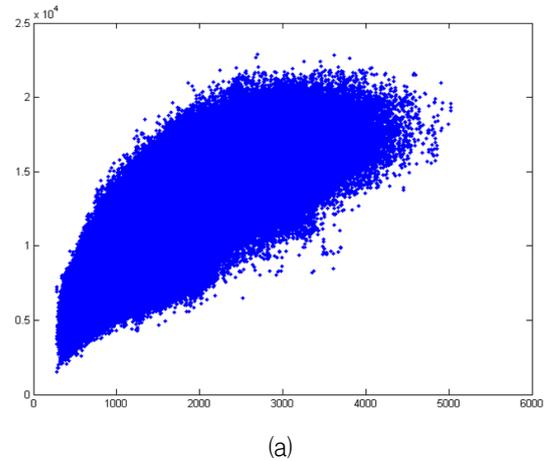
HEVC 인트라 예측의 연산 시간의 문제를 해결하기 위해서 고속 인트라 예측 방식에 대한 많은 연구가 이루어지고 있다. Jiang은 원본 픽셀의 기울기 정보를 이용하는 고속 인트라 모드 결정 방식을 제안하였다.<sup>[8]</sup> 이 방식은 기울기 정보를 이용하여 RDO 연산을 위한 후보 모드의 수를 줄임으로써 인코딩 속도를 향상 시켰다. Yan은 그룹 기반의 고속 모드 결정 알고리즘을 제안하였다.<sup>[9]</sup> 이 알고리즘은 RMD를 통해서 얻은 RDO 후보 모드들이 보통 인접한 모드일 가능성이 높다는 것을 발견하고, 이를 이용해 RDO 후보 모드의 수를 줄이

는 방식이다. Silva는 에지를 검출을 기반으로 하는 고속 인트라 모드 결정 방식을 제안하였다.<sup>[10]</sup> 이 방식은 4×4 블록 단위로 에지를 구하여 그 에지에 따라서 일부의 모드만 선택하여 인트라 예측을 진행하는 방식으로 고속 인트라 예측 방식을 구현하였다. 기존에 제안된 위의 세 가지 방법은 변환을 하는 경우와 TSM을 포함한 RDO를 위한 모드의 수를 줄임으로써 연산 시간을 감소시키는 방법으로 본 논문에서 제안하는 방식이 TSM에만 적용된다는 부분에서 차이가 있으며, 변환을 하는 부분과 TSM의 두 부분에 대해서 함께 적용되며 연산 시간을 많이 줄였다. 하지만 속도 향상에 따른 압축률 손실이 비교적 높게 나타나는 문제가 있다. 기존 고속 인트라 예측 방식 중에 속도 향상에 따른 압축률 손실을 최소화한 알고리즘으로는 본 연구 이전에 우리가 제안했던 고속 알고리즘인 SATD와 쿼드트리 구조를 이용한 고속 알고리즘이 있다.<sup>[11]</sup> 이 알고리즘은 기존 고속 알고리즘에 비해서 압축 효율 감소 비율 대비 높은 인코딩 시간 감소 비율을 가지고 있었으나, 인코딩 시간을 감소시키는 대신 감소하는 압축률이 매우 작아진 차이가 있을 뿐, 기존 고속 알고리즘과 같이 압축률 손실이 발생하게 된다. 본 논문에서는 이러한 기존 고속 인트라 예측 알고리즘과는 달리 압축률의 손실 없이 인트라 예측의 연산 시간을 감소시키는 새로운 방식을 제안한다.

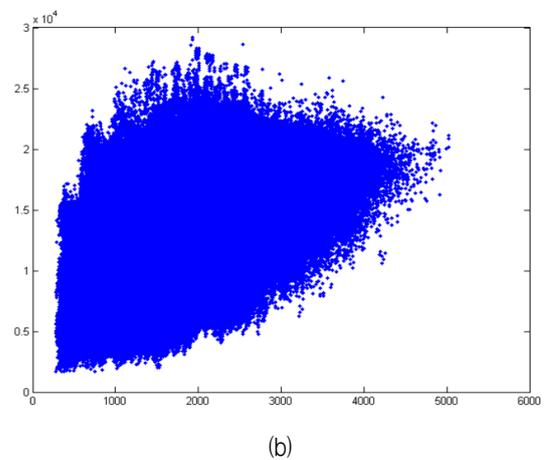
### III. 인트라 예측의 TSM 분석

일반적인 인코딩 처리 과정에서 원본 픽셀 값에서 예측된 픽셀 값을 빼고 남은 잔여 픽셀 값은 DCT나 DST를 이용하여 공간 도메인에서 주파수 도메인으로 변환 된다. 이때, 일반 영상의 잔여 픽셀 같은 경우에는 대부분의 데이터가 저주파 영역으로 압축되게 되고, 이것을 이용해서 압축이 진행되게 된다. 하지만 스크린 콘텐츠는 글씨 등에 의해서 날카로운 에지 부분이 많아서 일반 영상과는 달리 잔여 픽셀을 변환 할 경우 저주파 영역으로 압축되지 않는 데이터가 많게 되고, 이는 결국 변환 후 압축 과정에서 압축률 손실과 복원 영상의 화질 저하를 가져오게 된다.

TSM은 RDO 과정에서 DCT나 DST의 변환 과정을 생략해서 인코딩된 이미지의 모서리 부분이 손실되지 않도록 유지하며, 양자화로부터 데이터를 보호하기 위



(a)



(b)

그림 1. Hadamard 변환을 통한 비용과 RD 비용 사이의 상관관계

(a) DCT나 DST 변환을 통해 구한 RD 비용

(b) TSM을 통해 구한 RD 비용

Fig. 1. Correlation between the hadamard cost and the RD cost.

(a) the RD cost using DCT or DST transform

(b) the RD cost using TSM

해 스케일링을 한다. 이러한 TSM의 도입으로 HEVC는 스크린 콘텐츠의 코딩 효율을 비약적으로 향상시키게 되었다. HEVC의 참조 소프트웨어인 HM에서는 DCT나 DST 변환을 이용하는 RDO와 같이 hadamard 변환을 이용한 RMD를 통해서 TSM을 위한 후보 인트라 모드를 선택하여 TSM을 진행한다. 하지만 TSM 통해서 구한 RD 비용과 hadamard 변환을 이용한 비용 사이에 연관성이 매우 적어서 기존 HM의 인코더에서는 TSM의 성능이 저하되어 적용되어 있는 상황이다. 그림 1은 hadamard 변환을 이용한 비용과 RD 비용 사이의 상관관계를 나타낸다. 그림 1 (a)는 hadamard 변환

을 이용한 비용과 DCT나 DST 변환을 이용한 RD 비용 사이의 상관관계를 나타내며, 그림 1 (b)는 TSM을 이용한 RD 비용 사이의 상관관계를 나타낸다. 그림 1 (a)를 보면 두 비용의 관계 분포가 대각선의 선형 모양을 띠는 것을 볼 수 있다. 이것은 두 비용이 서로 높은 상관관계를 갖는다는 것을 나타낸다. 반면에 그림 1 (b)에서는 두 비용의 관계 분포가 넓게 퍼져서 분포되어 있는 것을 볼 수 있으며, 이것으로 두 비용의 상관관계가 낮다는 것을 확인 할 수 있다. 결국 기존의 hadamard 변환을 이용한 RMD를 이용해서는 TSM에 적절한 후보 모드를 선택하기 어렵다는 것을 보여준다. 따라서 본 논문에서는 TSM과 연관성이 높은 새로운 비용 함수를 적용하여 TSM의 성능을 극대화를 통한 압축 효율을 높이는 한편 임계값에 따른 TSM 과정의 생략으로 원래 RMD가 제안된 이유인 연산 시간 감소의 효과 역시 향상 시키는 TSM을 위한 새로운 RMD 알고리즘을 제안한다.

#### IV. 제안하는 방식

제안하는 RMD 알고리즘은 두 부분으로 구성된다. 먼저 첫 번째 부분에서는 새로운 비용 함수를 적용하여 적절한 후보 모드들을 선택하고, 이를 통해서 압축 효율을 향상시킨다. 다음으로 두 번째 부분에서는 새롭게 적용되는 임계값을 기반으로 불필요한 TSM 과정을 생략하여 인코딩 시간을 감소시킨다. 다음은 각 부분에 대한 세부적인 설명이다.

##### 1. 새로운 비용 함수 적용을 통한 압축률 향상을 위한 부분

HEVC의 기존 RMD 알고리즘에서는 hadamard 변환을 이용한다. 하지만 이 경우 hadamard 변환을 통해 구한 비용과 TSM 인코딩을 통해서 구한 RD 비용 사이의 낮은 상관관계로 인하여 TSM을 위한 최적의 예측 모드를 선택할 확률이 낮아지는 문제가 발생하게 된다. 이 문제를 해결하기 위해서 본 논문에서는 다음의 두 가지 조건을 만족시키는 새로운 비용 함수를 제안한다. 먼저 제안하는 비용 함수를 통해서 구한 비용은 TSM 인코딩을 통해서 구한 RD 비용과 높은 연관성을 가짐으로써 TSM을 위한 최적의 예측 모드를 높은 확률로 선택할 수 있는 조건을 만족 시켜야 한다.

다음으로 제안하는 비용 함수는 모든 인트라 예측 모드에 대해서 실행되어야 하고, 근본적으로는 연산 시간을 감소시키기 위하여 적용되는 함수이기 때문에 연산 시간을 최소화한 간단한 함수이어야 한다. 이와 같은 두 가지 조건을 고려하여 본 논문은 TSM을 통해서 구한 RD 비용과 비슷한 결과를 보이는 간단한 비용 함수를 제안한다.

HEVC의 기존 RMD는 DCT나 DST 변환을 대신하여 좀 더 간단한 변환인 hadamard 변환을 사용하여 연산 시간을 줄이는 방식을 사용한다. 즉, 변환을 하지 않는 TSM을 위한 RMD에서는 hadamard 변환을 사용할 필요가 없게 된다. 따라서 본 논문에서 제안하는 TSM을 위한 RMD의 새로운 비용 함수에서는 hadamard 변환을 하지 않는 순수한 SAD(sum of absolute difference) 값을 이용하여 비용을 구한다. 본 논문에서 제안하는 새로운 비용 함수는 다음과 같다.

$$Cost = \sum_{i=0}^N (|P_{org}[i] - P_{pred}[i]|) + \lambda(QP) \cdot Mode Bits \quad (1)$$

이때  $N$ 은 인트라 예측을 할 블록의 전체 픽셀의 수를 나타내며,  $P_{org}[i]$ 는 원본 픽셀 값을 나타내고,  $P_{pred}[i]$ 는 예측 픽셀 값을 나타낸다.  $\lambda(QP)$ 는 Lagrange multiplier를 나타내며,  $QP$ 는 양자화 파라미터를 나타내고,  $ModeBits$ 는 모드 코딩에 필요한 bit를 나타낸다.

새로운 비용 함수를 이용하여 제안하는 RMD는 비용

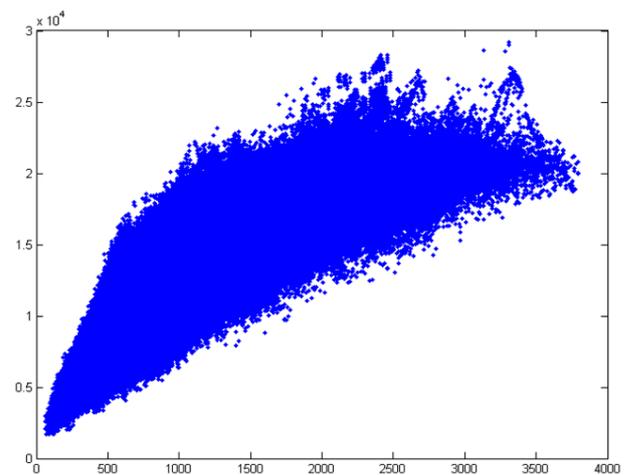


그림 2. 제안하는 비용 함수의 비용과 TSM을 통해 구한 RD 비용 사이의 상관관계

Fig. 2. Correlation between the cost using proposed cost function and the RD cost using TSM.

이 작은 예측 모드들로부터 TSM을 위한 후보 모드들을 선택하게 된다. 그림 2는 제안된 비용 함수로부터 구한 비용과 TSM을 통한 RD 비용 사이의 상관관계를 나타낸다. 그림 1 (b)의 기존 RMD에서 사용되는 hadamard 변환을 이용한 비용에 비해서 제안된 비용 함수의 결과가 TSM을 통한 RD 비용과 더욱 높은 상관관계를 가진다는 것을 확인할 수 있다. 이와 같은 높은 상관관계를 바탕으로 제안하는 RMD는 TSM을 위한 최적의 모드를 선택할 확률을 높이고 압축 효율을 향상 시키게 된다.

2. 임계값 도입을 통한 연산 시간 감소를 위한 부분

본 논문에서 제안하는 RMD 알고리즘은 임계값 도입을 통해 TSM 인코딩 과정을 생략하여 연산 시간을 감소시킨다. 우리는 실험을 통해서 만약 어느 하나의 인트라 예측 모드에 대한 TSM 인코딩 결과에서 모든 양자화 된 변환 계수(quantized transform coefficient)들이 0이 되는 경우, 모든 인트라 모드들에 대한 TSM을 통한 RD 비용에 비해서 DCT나 DST 변환을 통한 RD 비용이 작게 나타나고 결국 최종 인트라 예측 모드로 DCT나 DST 변환을 사용하는 결과가 선택 된다는 것을 발견할 수 있었다. 따라서 이러한 결과를 바탕으로 위에서 말한 조건을 만족시키는 경우에는 나머지 인트라 예측 모드들에 대한 TSM 과정을 생략하여 인코딩 시간을 줄일 수 있다. 이런 조건을 미리 확인하기 위해서 본 논문에서 제안하는 RMD에서는 임계값을 도입하였다. 양자화 된 변환 계수는 잔여 픽셀 값, 양자화 파라미터 그리고 비트 깊이(bit depth)에 의해서 결정된다. 이 사실에 따라 임계값은 아래 수식 (2)와 (3)과 같이 양자화 파라미터, 비트 깊이에 따라서 구한 후에 잔여 픽셀 값에 적용되어 사용하게 된다.

$$Threshold(bitdepth8) = 0.5654 \times \exp(0.1176 \times QP) \quad (2)$$

$$Threshold(bitdepth10) = 0.4802 \times \exp(0.1206 \times QP) \quad (3)$$

이때, QP는 양자화 파라미터를 나타내며, 수식 (2)는 비트 깊이가 8일 경우의 임계값이며, 수식 (3)은 비트 깊이가 10일 경우의 임계값을 나타낸다. 임계값의 두 수식들은 HEVC에서 사용되는 양자화 수식과 통계적인 실험 결과를 기초로 하여 유도된 결과이다.

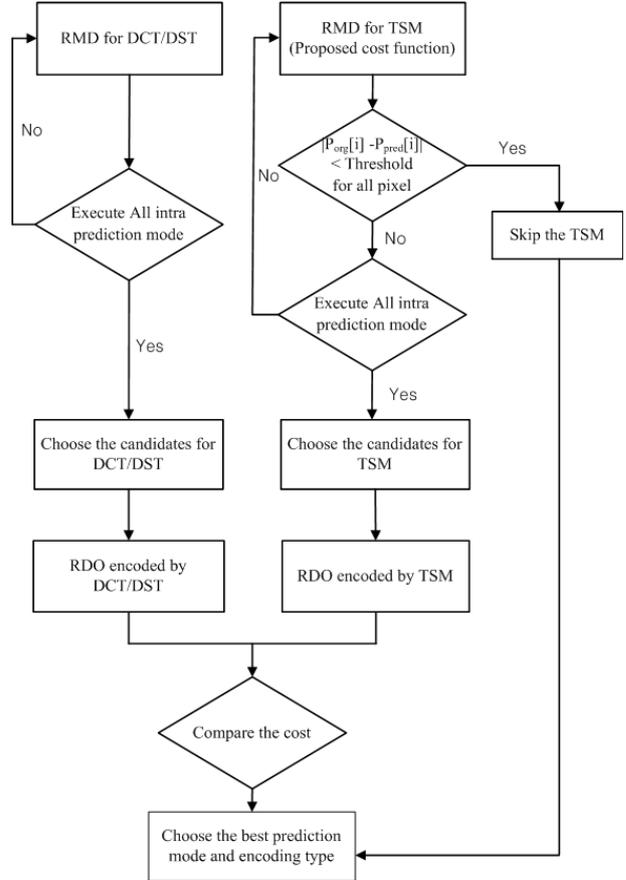


그림 3 인트라 예측 모드와 인코딩 타입을 결정하는 과정의 흐름도

Fig. 3 Flow chart of the process to determine the intra-prediction mode and encoding type.

그림 3은 본 논문에서 제안하는 RMD를 적용한 전체 시스템의 흐름도를 나타낸다. 흐름도를 보면 제안된 TSM을 위한 RMD가 기존 HEVC의 RMD와 함께 적용되어 있는 것을 확인할 수 있다. 제안된 RMD 과정을 보면 기존의 RMD와는 달리 TSM 생략 과정이 추가되어 있는 것을 볼 수 있다. 앞서 수식 (2)와 (3)에서 구한 임계값을 이용해서 각 인트라 예측 모드에서 잔여 픽셀 값들이 모두 임계값 보다 낮은 경우에는 즉시 TSM을 위한 과정을 생략하고 넘어가서 연산 시간을 단축시키게 된다.

V. 실험 결과

제안하는 고속 인트라 예측 방식의 검증을 위해서 HEVC 참조 소프트웨어인 HM 10.0<sup>[6]</sup>을 사용하였다. 실험 환경 구성은 공동 실험 조건문서<sup>[12]</sup>를 참조하였으며

자세한 내용은 표 1에 나타내었다. 제안하는 방식의 성능 검증을 위하여 우리는 제안하는 방식의 전체 인코딩 시간과 HM 10.0의 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 (4)와 같은 방식으로 계산하였다.

$$\Delta Time(\%) = \frac{(T_{pro} - T_{ref})}{T_{ref}} \quad (4)$$

식 (4)에서  $T_{pro}$ 는 제안하는 방식의 인코딩 시간이며,  $T_{ref}$ 는 HM 10.0의 인코딩 시간이다. 다음으로 압축 성능을 나타내기 위해서는 Bjontegaard-Delta(BD)<sup>[13]</sup> 측정법을 사용하였다. -x%의 BD-rate 값은 제안하는 방식이 기존 방식에 비해 x%의 전체 bit를 줄일 수 있다

표 1. 실험 조건 및 HM 구성

Table 1. 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) : Kimono, Park Scene, Basketball Drive, and Cactus</li> <li>• Class C (832×480) : Race Horses, BQ Mall, and Basketball Drill</li> <li>• Class D (416×240) : Race Horses, Blowing Bubbles, and Basketball Pass</li> <li>• Class E(1280×720) : Four People, Johnny, and KristenAndSara</li> <li>• Class F(screen contents) : Basketball Drill Text (832×480), China Speed(1024×768), Slide Editing(1280×720), and Slide Show(1280×720)</li> </ul>
영상의 실험 프레임 수	10초의 비디오 재생을 위한 프레임 수
소프트웨어	HM 10.0
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 2. 새로운 비용 함수를 적용한 방식의 인코딩 시간 및 압축 성능 (일반영상)

Table 2. Encoding time and compression performance of the method applying the new cost function. (general video).

Class	영상	QP	HM 10.0			제안한 방법 (new cost function)			성능 비교	
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	$\Delta Time$ (%)
A	Traffic	22	101878.91	43.35	1.31	101873.49	43.35	1.28	0.00	-2.87
		27	57384.08	40.16	1.13	57383.21	40.16	1.09		-3.93
		32	32738.70	37.12	0.98	32743.25	37.12	0.97		-1.53
		37	18511.60	34.10	0.89	18511.27	34.10	0.89		-0.48
	People On Street	22	104706.18	43.26	1.34	104718.24	43.26	1.30	0.00	-2.61
		27	60857.26	39.82	1.14	60855.69	39.82	1.11		-2.64
		32	34789.47	36.70	1.01	34783.29	36.70	1.01		-0.04
		37	20390.32	33.84	0.91	20390.02	33.84	0.90		-0.96
		37	20390.32	33.84	0.91	20390.02	33.84	0.90		-0.96
B	Kimono	22	22234.31	42.70	0.94	22231.30	42.70	0.92	0.00	-2.06
		27	12152.51	41.01	0.78	12152.87	41.01	0.76		-1.77
		32	6848.55	38.95	0.71	6847.95	38.95	0.73		1.76
		37	3820.53	36.51	0.66	3820.74	36.51	0.66		-0.93
	Park Scene	22	52736.77	41.68	1.16	52734.71	41.68	1.13	0.00	-2.54
		27	28591.58	38.60	0.94	28591.26	38.60	0.92		-1.89
		32	14837.37	35.61	0.80	14837.41	35.61	0.79		-1.88
		37	7300.55	32.79	0.71	7301.31	32.79	0.70		-1.11
		37	7300.55	32.79	0.71	7301.31	32.79	0.70		-1.11
	Cactus	22	105363.61	40.57	2.49	105370.06	40.57	2.44	-0.01	-2.13
		27	48710.15	37.93	1.94	48709.64	37.93	1.90		-1.65
		32	26370.06	35.69	1.64	26366.48	35.69	1.62		-1.21
37		14279.52	33.26	1.49	14280.57	33.26	1.47	-1.30		
Basketball Drive	22	71114.19	41.25	2.25	71109.34	41.25	2.21	0.00	-1.87	
	27	29165.12	38.71	1.77	29171.17	38.71	1.76		-0.60	
	32	15230.44	36.95	1.53	15231.15	36.95	1.51		-1.36	
	37	8495.71	34.98	1.42	8496.84	34.98	1.40		-1.49	
C	Basketball Drill	22	20808.93	41.76	0.48	20815.04	41.77	0.47	0.02	-2.66
		27	11313.41	38.37	0.39	11315.16	38.37	0.39		-1.32
		32	6075.09	35.41	0.34	6079.22	35.41	0.33		-1.46
		37	3355.56	32.77	0.30	3355.98	32.77	0.30		-1.20
	BQMall	22	23373.95	41.92	0.55	23374.89	41.92	0.54	0.00	-2.63
		27	13971.94	39.02	0.46	13973.63	39.02	0.45		-2.22
		32	8219.58	35.97	0.41	8221.78	35.97	0.40		-1.23
		37	4694.47	32.89	0.37	4695.40	32.89	0.36		-3.87
	Race Horses	22	14999.12	42.29	0.29	14999.76	42.29	0.28	0.00	-2.90
		27	9002.19	38.91	0.24	9003.64	38.91	0.24		-1.81
		32	5109.92	35.49	0.21	5108.94	35.49	0.21		-0.77
		37	2575.49	32.12	0.19	2575.68	32.12	0.18		-1.09
D	Basketball Pass	22	5321.44	42.95	0.11	5321.89	42.95	0.11	-0.02	-2.07
		27	3168.20	39.38	0.10	3167.04	39.38	0.09		-1.63
		32	1816.59	35.98	0.08	1816.80	35.99	0.08		-1.58
		37	1012.37	32.82	0.07	1012.69	32.82	0.07		-0.60
	Blowing Bubbles	22	11455.40	41.03	0.15	11456.30	41.03	0.14	-0.02	-3.07
		27	7073.62	36.72	0.13	7071.74	36.72	0.12		-4.22
		32	4023.30	32.74	0.11	4023.69	32.74	0.10		-2.71
		37	2073.17	29.21	0.09	2073.36	29.21	0.09		-1.35
	Race Horses	22	4459.94	42.42	0.07	4460.44	42.42	0.07	0.00	-2.61
		27	2684.48	38.41	0.06	2684.04	38.41	0.06		-2.02
		32	1475.23	34.58	0.05	1474.93	34.58	0.05		-0.93
		37	744.35	31.29	0.05	744.29	31.28	0.05		-0.81
E	Four People	22	30227.07	43.80	1.05	30228.98	43.80	1.03	0.00	-1.68
		27	18538.75	41.23	0.92	18538.05	41.23	0.91		-1.40
		32	11401.79	38.36	0.84	11401.54	38.36	0.83		-0.88
		37	6930.59	35.27	0.78	6930.63	35.27	0.78		-0.59
	Johnny	22	20142.91	44.05	0.97	20143.44	44.05	0.95	0.00	-2.65
		27	11233.27	41.84	0.84	11233.77	41.84	0.84		-0.37
		32	6423.21	39.50	0.78	6423.34	39.50	0.77		-0.90
		37	3708.33	36.90	0.73	3708.33	36.90	0.73		-0.39
	KristenAndSara	22	22193.82	44.42	0.97	22191.30	44.42	0.95	0.00	-1.71
		27	13262.08	42.10	0.85	13261.38	42.10	0.85		-0.95
		32	8032.71	39.46	0.79	8033.08	39.46	0.79		-0.49
		37	4860.67	36.57	0.75	4860.45	36.57	0.75		-0.38
평균									0.00	-1.60

는 것을 의미하고 반대로 양수 값을 갖는 경우에는 전체 bit를 증가시켜 압축 효율이 떨어졌다는 것을 의미한다. HM 10.0과 비교한 제안하는 방식의 최종 성능은 표 2~5에 나타내었다. 먼저 표 2는 IV장 1절에서 설명한 압축 성능 향상을 위한 새로운 비용 함수를 적용한 방식의 일반 영상에서의 인코딩 시간 감소 비율 및 압축 성능을 나타내며, 표 3은 이 방식의 스크린 콘텐츠에서의 성능을 나타낸다. 표 2를 보면 일반 영상에서 제안하는 방식을 통한 특별한 성능 변화가 없는 것을 알 수 있다. 이는 제안하는 방식이 적용되는 TSM 자체가 일반 영상보다는 스크린 콘텐츠에서의 성능 향상을 위한 알고리즘이기 때문에 제안하는 방식이 적용되어도 일반 영상에서는 크게 성능 변화가 없게 된다. 제안하는 방식의 압축 성능은 표 3을 통해서 확인 할 수 있다. 표 3을 보면 제안하는 방식이 스크린 콘텐츠에서 인코딩 시간의 손실 없이 평균 0.37%의 압축률을 향상시키는 것을 볼 수 있다. 즉, 두 표의 결과를 종합해 볼 때, 본 논문에서 제안하는 비용 함수가 적용될 경우 기존 HM 10.0에 비해서 모든 영상에 대해서 인코딩 시간의 손실이 없고, 스크린 콘텐츠에 대해서는 압축률을 향상시킬 수 있다는 것을 확인할 수 있다.

표 4와 표 5에서는 IV장 1절에서 설명한 새로운 비용 함수를 적용하고, IV장 2절에서 설명한 임계값 도입을

표 3. 새로운 비용 함수를 적용한 방식의 인코딩 시간 및 압축 성능 (스크린 콘텐츠)

Table 3. Encoding time and compression performance of the method applying the new cost function. (screen contents).

Class	영상	QP	HM 10.0			제안한 방법 (new cost function)			성능 비교	
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	ΔTime (%)
F	BasketballDrillText	22	22727.02	41.98	0.49	22729.81	41.99	0.48	-0.05	-2.26
		27	13152.02	38.53	0.41	13150.71	38.53	0.40		-1.57
		32	7612.66	35.43	0.35	7612.63	35.43	0.35		-1.86
		37	4476.68	32.53	0.31	4477.26	32.53	0.31		-0.57
	ChinaSpeed	22	22776.51	44.90	0.86	22698.51	44.92	0.84	-0.68	-1.51
		27	15172.07	40.97	0.76	15117.61	41.00	0.75		-1.17
		32	9879.63	37.20	0.69	9840.22	37.23	0.68		-0.85
		37	6433.37	33.61	0.63	6406.88	33.64	0.62		-1.11
	SlideEditing	22	33420.98	46.61	0.65	33335.06	46.65	0.64	-0.66	-1.45
		27	24988.22	42.25	0.61	24924.79	42.31	0.60		-1.30
		32	19055.21	37.72	0.57	18988.41	37.78	0.56		-1.60
		37	14253.42	32.95	0.52	14223.97	33.02	0.51		-1.41
SlideShow	22	5180.96	50.78	0.70	5179.07	50.79	0.70	-0.09	-0.55	
	27	3525.72	47.14	0.66	3522.38	47.14	0.66		-0.77	
	32	2418.94	43.47	0.64	2417.65	43.47	0.64		-0.75	
	37	1623.61	39.49	0.61	1623.56	39.49	0.61		-0.57	
평균									-0.37	-1.21

표 4. 새로운 비용 함수와 고속 알고리즘을 함께 적용한 방식의 인코딩 시간 및 압축 성능 (일반 영상)

Table 4. Encoding time and compression performance of the proposed method applying the new cost function and fast algorithm. (general video).

Class	영상	QP	HM 10.0			제안한 방법 (new cost function + fast)			성능 비교		
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	ΔTime (%)	
A	Traffic	22	101878.91	43.35	1.31	101883.48	43.35	1.19	0.01	-9.26	
		27	57384.08	40.16	1.11	57382.13	40.16	0.99		-11.02	
		32	32738.70	37.12	0.98	32742.40	37.12	0.89		-8.93	
		37	18511.60	34.10	0.89	18509.31	34.10	0.78		-12.60	
	PeopleOnStreet	22	104706.18	43.26	1.33	104686.83	43.26	1.22	0.00	-8.36	
		27	60857.26	39.82	1.14	60838.20	39.82	1.02		-10.16	
		32	34789.47	36.70	1.00	34784.11	36.70	0.88		-12.04	
		37	20390.32	33.84	0.91	20391.73	33.84	0.83		-8.86	
	B	Kimono	22	22234.31	42.70	0.94	22230.49	42.70	0.83	0.00	-11.68
			27	12152.51	41.01	0.78	12152.84	41.01	0.69		-11.61
			32	6848.55	38.95	0.71	6848.46	38.95	0.62		-12.55
			37	3820.53	36.51	0.66	3821.15	36.51	0.58		-12.38
ParkScene		22	52736.77	41.68	1.16	52727.45	41.68	1.06	0.01	-8.31	
		27	28591.58	38.60	0.95	28583.28	38.60	0.84		-11.41	
		32	14837.37	35.61	0.80	14836.02	35.61	0.70		-11.50	
		37	7300.55	32.79	0.71	7300.41	32.79	0.62		-12.31	
Cactus		22	105363.61	40.57	2.50	105301.67	40.57	2.31	0.01	-7.92	
		27	48710.15	37.93	1.94	48694.48	37.93	1.74		-10.47	
		32	26370.06	35.69	1.64	26360.67	35.69	1.44		-11.99	
		37	14279.52	33.26	1.48	14279.64	33.26	1.31		-12.07	
BasketballDrive	22	71114.19	41.25	2.26	71090.88	41.25	2.06	0.01	-8.64		
	27	29165.12	38.71	1.76	29154.88	38.71	1.56		-11.25		
	32	15230.44	36.95	1.53	15230.22	36.95	1.40		-9.00		
	37	8495.71	34.98	1.42	8496.82	34.98	1.27		-10.62		
C	BasketballDrill	22	20808.93	41.76	0.48	20803.09	41.76	0.44	0.00	-7.98	
		27	11313.41	38.37	0.39	11307.96	38.37	0.36		-9.59	
		32	6075.09	35.41	0.34	6073.88	35.41	0.31		-7.60	
		37	3355.56	32.77	0.30	3355.13	32.77	0.27		-11.19	
	BQMall	22	23373.95	41.92	0.55	23371.54	41.92	0.51	0.01	-8.05	
		27	13971.94	39.02	0.46	13970.18	39.02	0.42		-9.81	
		32	8219.58	35.97	0.41	8219.02	35.96	0.36		-10.79	
		37	4694.47	32.89	0.37	4692.13	32.88	0.32		-12.97	
	RaceHorses	22	14999.12	42.29	0.29	14998.39	42.29	0.26	-0.01	-7.93	
		27	9002.19	38.91	0.24	8999.70	38.91	0.22		-8.94	
		32	5109.92	35.49	0.21	5106.02	35.49	0.19		-11.33	
		37	2575.49	32.12	0.19	2572.71	32.11	0.16		-12.04	
D	BasketballPass	22	5321.44	42.95	0.11	5320.74	42.95	0.10	-0.01	-10.46	
		27	3168.20	39.38	0.10	3166.96	39.38	0.09		-12.18	
		32	1816.59	35.98	0.08	1815.58	35.98	0.07		-12.18	
		37	1012.37	32.82	0.07	1012.42	32.81	0.07		-11.29	
	BlowingBubbles	22	11455.40	41.03	0.15	11452.62	41.03	0.14	0.00	-4.61	
		27	7073.62	36.72	0.13	7068.43	36.71	0.12		-6.03	
		32	4023.30	32.74	0.11	4019.83	32.73	0.10		-9.86	
		37	2073.17	29.21	0.09	2068.59	29.20	0.08		-12.06	
	RaceHorses	22	4459.94	42.42	0.07	4460.46	42.42	0.07	0.01	-6.22	
		27	2684.48	38.41	0.06	2684.67	38.41	0.06		-8.44	
		32	1475.23	34.58	0.05	1475.04	34.58	0.05		-11.56	
		37	744.35	31.29	0.05	744.28	31.28	0.04		-11.58	
E	FourPeople	22	30227.07	43.80	1.05	30229.41	43.80	0.96	-0.01	-8.63	
		27	18538.75	41.23	0.92	18533.57	41.23	0.83		-10.39	
		32	11401.79	38.36	0.84	11399.90	38.36	0.74		-11.55	
		37	6930.59	35.27	0.78	6929.19	35.27	0.69		-11.85	
	Johnny	22	20142.91	44.05	0.97	20142.23	44.05	0.87	0.00	-10.70	
		27	11233.27	41.84	0.84	11231.88	41.84	0.74		-11.62	
		32	6423.21	39.50	0.78	6422.42	39.50	0.68		-12.15	
		37	3708.33	36.90	0.74	3706.38	36.89	0.65		-12.52	
	KristenAndSara	22	22193.82	44.42	0.99	22191.72	44.42	0.87	0.00	-12.06	
		27	13262.08	42.10	0.88	13260.32	42.10	0.76		-13.25	
		32	8032.71	39.46	0.81	8031.70	39.46	0.70		-13.22	
		37	4860.67	36.57	0.76	4860.24	36.57	0.66		-12.46	
평균									0.00	-10.47	

통한 연산 시간 감소 알고리즘을 함께 적용한 최종 제안 방식의 성능을 나타낸다. 표 4는 최종 제안 방식의 일반 영상에서의 성능을 나타낸다. 표 4를 보면 표 2의 결과와는 달리 고속 알고리즘의 적용으로 인코딩 시간이 줄어든 것을 확인 할 수 있다. 앞서 표 2에서 제안한 새로운 비용 함수가 TSM에 적용되고 TSM이 스크린 콘텐츠의 압축률을 향상을 시키기 위한 알고리즘이기 때문에 압축률 향상에 특별한 영향이 없다고 설명했다. 표 4에서도 마찬가지로 이유로 새로운 비용 함수 적용에 따른 압축률 향상이 없으며, 임계값 도입을 통한 연산 시간 감소 알고리즘 역시 TSM에 적용되기 때문에 고속 알고리즘 적용에 따른 압축률 손실 또한 없다. 즉, 일반 영상에서는 제안하는 방식의 적용에 따른 압축률의 변화가 나타나지 않게 된다. 반면 인코딩 시간 부분에서는 제안하는 고속 알고리즘을 통해서 인코딩 과정에서 일부 TSM 과정을 생략하여 인코딩 시간을 감소 시키게 된다. 표 4의 결과를 보면 인코딩 시간이 전체 영상에 대해서 평균 10.47%가 감소된 것을 확인 할 수 있다. QP에 따른 인코딩 시간 감소량을 보면 QP가 증가할수록 인코딩 시간 감소량이 증가하는 것을 확인 할 수 있다. QP 22에서 평균 8.72%, QP 27에서 평균 10.30%, QP 32에서 평균 11.08%, QP 37에서 평균

표 5. 새로운 비용 함수와 고속 알고리즘을 함께 적용한 방식의 인코딩 시간 및 압축 성능 (스크린 콘텐츠)

Table 5. Encoding time and compression performance of the proposed method applying the new cost function and fast algorithm. (screen contents).

Class	영상	QP	HM 10.0			제안한 방법 (new cost function + fast)			성능 비교	
			Bitrate (kbps)	PSNR (dB)	T (h)	Bitrate (kbps)	PSNR (dB)	T (h)	BD-rate (%)	ΔTime (%)
F	Basketball	22	22727.02	41.98	0.49	22720.66	41.98	0.45	-0.04	-7.67
		27	13152.02	38.53	0.41	13143.07	38.52	0.37		-9.14
		32	7612.66	35.43	0.35	7608.14	35.43	0.31		-11.39
		37	4476.68	32.53	0.32	4470.70	32.52	0.28		-13.70
	DrillText	22	22776.51	44.90	0.86	22680.03	44.91	0.79	-0.50	-8.12
		27	15172.07	40.97	0.77	15085.12	40.96	0.70		-8.70
		32	9879.63	37.20	0.69	9802.82	37.18	0.63		-9.58
		37	6433.37	33.61	0.63	6360.49	33.54	0.56		-10.36
	ChinaSpeed	22	33420.98	46.61	0.65	33334.80	46.65	0.60	-0.57	-8.11
		27	24988.22	42.25	0.61	24915.53	42.30	0.56		-8.60
		32	19065.21	37.72	0.57	18964.88	37.74	0.52		-8.42
		37	14253.42	32.95	0.53	14160.63	32.91	0.48		-10.16
SlideEditing	22	5180.96	50.78	0.70	5177.60	50.79	0.62	-0.07	-11.62	
	27	3525.72	47.14	0.67	3521.46	47.13	0.59		-12.17	
	32	2418.94	43.47	0.64	2413.55	43.46	0.56		-11.76	
	37	1623.61	39.49	0.61	1619.65	39.47	0.54		-12.39	
SlideShow										
평균								-0.30	-10.12	

11.79%의 인코딩 시간 감소되었다. 이처럼 QP가 증가할수록 인코딩 시간이 감소하는 것은 제안하는 고속 알고리즘이 적용되는 TSM의 특성 때문에 발생하는 현상이다. 앞서 이야기한 것과 같이 TSM은 스크린 콘텐츠의 성능 향상을 위해서 제안된 알고리즘이다. 보통 스크린 콘텐츠는 영상에 모서리 부분이 많다는 특성을 가지고 있으며, TSM은 이런 모서리 부분의 압축에 특화되어, 기존 코덱 알고리즘이 영상의 저주파 부분의 압축에 초점을 맞추고 있다면 TSM은 영상의 고주파 부분을 압축 하는데 초점이 맞추어져 있다. 즉, 영상에 모서리 부분의 고주파 성분이 많게 되면 TSM이 많이 사용되게 되고 반대의 경우에는 거의 사용되지 않게 된다. QP에 따른 영상을 보면, 같은 영상일 때 QP가 커질수록 영상의 모서리 부분이 뭉개지면서 영상의 고주파 성분이 많이 없어지게 된다. 따라서 QP가 커지면 QP가 작을 때보다 상대적으로 TSM이 사용될 경우가 더 줄어들게 된다. 제안하는 알고리즘이 최종 모드로 선택될 가능성이 낮은 TSM의 인코딩 과정을 생략하게 되는데 QP가 큰 영상일수록 제안하는 알고리즘을 통한 TSM 과정이 많이 생략되면서 인코딩 시간 감소 비율이 더욱 증가하게 된다. 표 5에서는 최종 제안방식의 스크린 콘텐츠에서의 성능을 나타낸다. 제안하는 알고리즘은 스크린 콘텐츠에서 평균 0.3%의 압축률을 향상 시키며, 평균 10.12%의 인코딩 시간을 감소시키며, 압축률과 인코딩 시간을 함께 줄이는 좋은 성능을 보여준다. 표 3의 결과와 비교를 해보면 새로운 비용 함수만 적용한 결과에 비해서 고속 알고리즘을 함께 적용한 결과 압축 효율은 평균 0.07%로 매우 약간 떨어진 반면에 인코딩 시간 감소율은 평균 약 9%정도가 증가하는 성능을 얻

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
	Jiang's	Yan's	Silva's	
평균 인코딩 시간 감소(%)	19.99	23.52	18.88	10.12
평균 압축 성능 감소(%)	0.74	1.3	1.30	0.00 (일반영상) -0.3 (스크린 콘텐츠)

은 것을 확인할 수 있다.

표 6은 II장에서 소개한 기존의 고속 인트라 예측 방식과 제안 방식의 성능을 비교하여 나타낸 것이다. 평균 인코딩 시간 감소 부분에서는 상대적으로 제안 방식의 성능이 낮지만 기존 방식들은 인코딩 시간 감소를 위해서 압축 성능 부분에서 손실이 발생하는 반면 제안하는 방식은 일반 영상에서는 압축 성능의 손실이 없고, 스크린 콘텐츠에서는 오히려 압축 성능을 향상시키는 것을 확인할 수 있다. 제안하는 방식은 일반 영상에 대해서는 추가적인 성능 손실 없이 인코딩 시간을 감소시키고, 특히 스크린 콘텐츠에서는 추가적으로 압축 효율도 향상시키는 제안 방식이 기존 고속 인트라 예측 방식들에 비해서 좀 더 효율적으로 사용될 수 있는 알고리즘이라는 것을 확인할 수 있다.

#### IV. 결 론

본 논문에서는 HEVC의 TSM를 위한 RMD 알고리즘을 제안하였다. 제안 방식은 TSM의 RMD를 위해서 새로운 비용 함수를 적용하여 최선의 TSM 선택 가능성을 높이고 압축 효율을 향상 시켰으며, 임계값을 이용하여 불필요한 TSM 선택 과정을 생략하여 인코딩 시간을 감소시켰다. 실험 결과 제안하는 알고리즘은 일반 영상에서는 압축 성능 손실 없이 평균 10%이상의 인코딩 시간을 감소시켰으며, 스크린 콘텐츠에서는 평균 0.3%의 압축 성능 향상과 함께 평균 10%이상의 인코딩 시간을 감소시켰다. 이 실험 결과는 본 논문에서 제안하는 알고리즘이 실시간 인코딩을 위한 고속 인트라 예측 방식의 좋은 대안이 될 수 있음을 보여준다.

#### REFERENCES

- [1] 심동규, 조현호 “고효율 영상 부호화 기술 HEVC 표준 기술의 이해”, 홍릉과학출판사, 1쪽-442쪽, 2014년
- [2] 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.
- [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] M. Mark, A. Gabriellini, and D. Flynn, “Transform skip mode,” presented at JCTVC-G575, *JCT-VC of ISO/IEC and ITU-T*, Geneva, CH, Nov. 2011.
- [5] 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.
- [6] HEVC Test Model 10 “HM 10.0”, [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-10.0](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-10.0)
- [7] Y. Kim, D. Jun, S. Jung, and J. Choi, “A fast intra prediction method using hadamard transform in high efficiency video coding”, *Proc. SPIE8305, Visual Information Processing and Communication III*, vol. 83050A, pp.1-10, San Francisco, Feb. 2012.
- [8] W. Jiang, H. Ma, and Y. Chen, “Gradient based fast mode decision algorithm for intra prediction in HEVC”, *Proc. Consumer Electronics, Communications and Networks (CECNet)* pp.1836-1840, Hubei, China, Apr. 2012.
- [9] S. Yan, L. Hong, W. He and Q. Wang, “Group-based fast mode decision algorithm for intra prediction in HEVC”, *Proc. Signal Image Technology and Internet Based Systems (SITIS)*, pp. 225-229, Kyoto, Japan, Nov. 2012.
- [10] 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.
- [11] Youngjo Kim, and Jaeseok Kim, “A Fast Intra Prediction Method Using Quadtree Structure and SATD in HEVC Encoder”, *Journal of The Institute of Electronics and Information Engineers*, Vol. 51, Issue 3, pp. 129-138, 2014
- [12] 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.
- [13] G. Bjøntegaard, “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 책임 연구원.  
1995년~현재 연세대학교 전기전자공학과 교수.  
<주관심분야: 통신 및 영상 시스템, VLSI 신호  
처리, 임베디드 S/W 및 SoC 구현>