

일반논문 (Regular Paper)

방송공학회논문지 제21권 제4호, 2016년 7월 (JBE Vol. 21, No. 4, July 2016)

<http://dx.doi.org/10.5909/JBE.2016.21.4.578>

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

움직임 벡터의 계층적 군집화를 통한 HEVC 고속 부호화 연구

임 정 윤, 안 용 조, 심 동 규^{a)†}

Study on Fast HEVC Encoding with Hierarchical Motion Vector Clustering

Jeongyun Lim^{a)}, Yong-Jo Ahn^{a)}, and Donggyu Sim^{a)†}

요 약

본 논문에서는 HEVC 부호화기의 속도를 향상시키기 위하여, 움직임 벡터의 군집화를 통한 코딩 블록의 최대 분할 깊이를 결정하는 방법을 제안한다. 현재 HEVC (High Efficiency Video Coding)의 참조 소프트웨어 HM은 최적의 코딩 블록 구조를 찾기 위해 다양한 코딩 블록의 깊이들에 대한 율-왜곡 최적화 (RDO: Rate-Distortion Optimization)를 수행한다. 하지만 이는 부호화기의 높은 복잡도를 차지하는 요소 중 하나로 보고된다. 본 논문에서는 최적의 코딩 블록 구조를 움직임 벡터의 군집화 된 결과에 따라 결정함으로써, 부호화 과정에서 코딩 블록 구조를 찾기 위한 RDO의 복잡도를 줄임으로써 부호화기의 속도를 향상시키는 방법을 제안한다. 제안하는 방법은 전처리 과정에서 원본 영상에 대한 움직임 벡터 계산, 이를 통한 계층적 군집화를 수행하여, 군집화된 경향을 기반으로 코딩 블록의 최대 깊이를 결정한다. 본 논문의 제안하는 방법은 HEVC 참조 소프트웨어 대비 평균 1.45% BD-rate 손실이 있었으며 평균 16%의 부호화 속도 향상을 보였다. 또한, 기존의 고속화 방법과 함께 적용한 경우 1.84% BD-rate 손실과 45.13%의 평균 부호화 속도 향상을 나타냈다.

Abstract

In this paper, the fast encoding algorithm in High Efficiency Video Coding (HEVC) encoder was studied. For the encoding efficiency, the current HEVC reference software is divided the input image into Coding Tree Unit (CTU). then, it should be re-divided into CU up to maximum depth in form of quad-tree for RDO (Rate-Distortion Optimization) in encoding process. But, it is one of the reason why complexity is high in the encoding process. In this paper, to reduce the high complexity in the encoding process, it proposed the method by determining the maximum depth of the CU using a hierarchical clustering at the pre-processing. The hierarchical clustering results represented an average combination of motion vectors (MV) on neighboring blocks. Experimental results showed that the proposed method could achieve an average of 16% time saving with minimal BD-rate loss at 1080p video resolution. When combined the previous fast algorithm, the proposed method could achieve an average 45.13% time saving with 1.84% BD-rate loss.

Keyword : HEVC, Clustering, Motion vector, Fast encoding

a) 광운대학교 컴퓨터공학과(Dept. of Computer Engineering, Kwangwoon University)

† Corresponding Author : 심동규(Donggyu Sim)

E-mail: dgsim@kw.ac.kr

Tel: +82-2-941-6470

ORCID: <http://orcid.org/0000-0002-2794-9932>

※ 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2014R1A2A1A11052210)과 2014년도 정부(미래창조과학부)의 재원으로 연구성과실용화진흥원의 지원을 받아 수행된 기초연구성과활용지원사업(연구성과사업지원사업).

· Manuscript received April 25, 2016; Revised July 18, 2016; Accepted July 22, 2016.

1. 서론

소비자의 고화질 및 4K, 8K-UHD (Ultra High Definition)과 같은 고해상도 영상에 대한 요구가 높아짐에 따라, MPEG-2 또는 H.264/AVC와 같은 종래의 대표적인 비디오 부호화 표준을 이용한 영상 부호화 서비스 한계에 도달하였다. 이에 고효율, 고해상도를 위한 비디오 부호화 기술의 필요성이 증대되었고, 2010년 ISO/IEC MPEG (Moving Picture Expert Group)과 ITU-T VCEG (Video Coding Expert Group)은 공동으로 JCT-VC (Joint Collaborative Team on Video Coding)를 결성, 2013년 1월 최신 비디오 표준인 HEVC (High Efficiency Video Coding) version 1 표준 기술 개발을 완료하였다^[1]. 이후, version 1을 기반으로 다양한 응용에서 사용할 수 있는 확장 표준을 진행하였으며, SHVC (HEVC Scalable Extension), MV-HEVC (HEVC Multi-view Extension), HEVC RExt (Range Extension)을 포함하는 version 2 표준을 공표하였다.

HEVC는 최신 비디오 표준으로 멀티미디어 관련 시장에서 다양한 응용에서 종래의 비디오 표준을 대체하여 사용될 것으로 기대되고 있다. 더욱이, HEVC는 H.264/AVC 대비 주관적 화질 측면에서 약 50%의 비트율 감소, 객관적 화질 측면에서 약 40%의 비트율 감소를 보이면서 높은 부호화 효율 보이는 것으로 보고되었다. 하지만 H.264/AVC 대비 다양해진 코딩 블록 구조와 확장된 화면 내 예측 모드의 방향, 화면 간 예측 모드의 정밀도 향상 및 인 루프 필터 (In-loop filter) 단계에서의 샘플 적응적 오프셋 (SAO: Sample Adaptive Offset) 모듈 추가 등으로 인해 부호화 과정에서의 부호화 복잡도 역시 크게 증가하였다^[2]. 이러한 문제점을 보완하고자, HEVC 부호화기의 복잡도를 낮추기 위한 연구가 활발히 진행되고 있다.

HEVC 부호화기의 고속화를 위한 접근 방법 중 하나는 코딩 구조를 가지기 (Pruning)를 통하여 부호화 속도를 향상시키는 방법이 있다. 화면 간 예측 모드를 통한 부호화 방법으로는 최상위 블록에 대한 부호화를 먼저 수행하고, 움직임 벡터 및 잔차 신호등을 확인하여 하위 블록으로의 분할을 수행하지 않는 부호화 방법^[4]이 있다. 이와 유사한 접근 방법으로는 상위 블록에서 특정 부호화 모드인

SKIP Mode로 부호화 될 경우, 하위 블록으로 분할하지 않는 부호화 방법 HEVC 표준화 과정에서 제안되어 참조 소프트웨어에 채택되었다^[5]. 이러한 연구들은 상위 블록에서 하위 블록으로 분할하며 부호화를 수행하는 HEVC 부호화 코딩 블록 구조를 확률적, 실험적으로 분석하여, 상위 블록에서 부호화 정보를 기반으로 하위 블록으로 계층적 분할하기 이전에 조기 종료 시키는 고속 부호화 방법이다. 다른 접근 방식의 고속화 연구로는 HEVC 부호화 과정에서 모듈별 고속화 알고리즘이 있다. 대표적인 방법은 부호화 과정에서 가장 높은 복잡도를 차지하는 화면 간 예측에서의 복잡도를 줄이는 방법으로써, 주로 움직임 추정 단계에서의 조기 종료 알고리즘^{[6][7]}들이 연구되었다. 또한, 부호화 과정을 병렬 처리를 통해 고속으로 처리하는 연구가 수행되었는데, 데이터 종속성이 없는 영역에 대해 병렬화 기법인 SIMD 구현을 통한 고속 부호화 연구^[8]가 발표되었다. 이 외에도 부호화 과정 외의 별도의 전처리 과정을 수행하여 고속으로 부호화를 수행하는 방법이 발표되었다^[9]. 이는 부호화를 수행하기 이전에, 전처리 과정에서 원본 입력 영상에 움직임 추정 및 화면 내 예측을 수행하여 부호화에 필요한 정보를 획득하는 알고리즘이다. 움직임 추정을 통해 획득한 움직임 벡터 (Motion vector) 및 참조 픽처 (Reference picture) 등과 같은 움직임 정보를 활용하여 부호화 과정에서 블록의 예측 방향 (단방향 예측 (Uni-prediction) 또는 양방향 예측 (Bi-prediction))을 미리 결정하여 부호화 속도를 향상시킨다. 또한, 화면 내 예측 정보와 움직임 추정 정보를 이용하여 부호화를 수행하는 블록이 앞으로 얼마만큼 참조되는지를 가늠하여, Rate control을 위한 QP 조절 및 비트를 할당을 수행한다.

본 논문에서는 전처리 과정에서 움직임 벡터들에 대한 계층적 군집화를 수행하여 HEVC 부호화 속도를 향상시키는 방법을 제안한다. 현재 HEVC 참조 소프트웨어에서는 부호화 효율을 고려하여, 입력 영상을 코딩 블록 단위로 분할, 코딩 블록은 쿼드 트리 형태의 하향식 계층 분할을 재귀적으로 수행하며, 순차적으로 부호화를 수행한다. 이러한 하향식 계층적 부호화 방식은 부호화 효율을 높일 수 있지만, 부호화 과정에서의 큰 복잡도를 차지하는 원인 중 하나이다. 본 논문은 부호화 속도를 저하시키는 하향식 계층적

분할 방법을 해결하고자, 전처리 과정에서 움직임 추정 단계를 통해 움직임 벡터를 지표로 하는 계층적 군집화 (Hierarchical Clustering)를 수행한다. 전처리 과정에서 최소 CU 크기에 해당하는 블록들의 계층적 군집화를 통해 군집화 정도를 나타내는 군집화 비용을 계산하며, 이는 블록들의 군집 정도를 나타낸다. 본 논문은 전처리 과정에서 계산한 블록들의 군집 정도를 이용하여, 부호화 과정에서의 코딩 블록의 깊이를 결정함으로써 부호화기의 속도를 향상시키는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 관련 연구로써, 기존 HEVC 고속 부호화 방법에 대해 Pruning 기반의 고속 부호화 알고리즘과 전처리 기반의 고속 부호화 알고리즘에 대해 설명한다. 3장에서는 본 논문에서 제안하는 전처리 기반의 HEVC 고속 부호화 알고리즘에 대해 설명한다. 4장에서는 본 논문에서 제안하는 알고리즘에 대한 실험 결과와 성능을 분석한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

II. 기존의 HEVC 고속 부호화 연구

본 논문의 2장에서는 기존의 HEVC 부호화기의 부호화 속도를 향상시키기 위한 방법을 Pruning 기반의 고속화 방법과

전처리 기반의 부호화 방법으로 분류하여 소개한다. 먼저, 2.1에서는 Pruning 기반의 고속화 방법은 HEVC의 계층적 코딩 블록 구조에서 상위 블록의 코딩 블록의 부호화 경향을 기반으로 하위 블록으로의 분할 과정을 생략하는 고속 부호화 방법을 설명하며, 2.2에서는 전처리 기반의 부호화 속도 및 부호화 효율 향상에 대한 기존 연구에 대해 설명한다.

1. Pruning 기반의 고속 부호화 연구

HEVC 참조 소프트웨어인 HM은 부호화 효율을 향상시키기 위해, 최적의 CU 크기를 결정하기 위한 RDO 과정을 수행한다. 이러한 RDO 과정은 일반적으로 CU는 최소 깊이 (Minimum depth: 0)부터 최대 깊이 (Maximum depth: 3)까지 쿼드 트리 형태로 분할하며 예측과 변환을 수행하게 되는데, 최소 깊이의 CU0는 64X64 크기를 가지며 해당 크기에서 PU와 TU로 분할 예측과 변환을 수행한다. 이후 해당 CU0를 4개의 32X32로 분할하여 예측과 변환을 수행한다. 이러한 과정은 최적의 부호화 모드 찾기 위하여 최대 깊이가 3까지 수행하며, 최소 깊이가 0부터 최대 깊이가 3까지 가변 크기를 갖는 CU에 대한 계산 복잡도는 식 (1)과 같다^[12].

$$\begin{aligned} f(n) &= f(n-1) + 4^n * M_n, f(0) \\ &= M_0 \text{ and } M_i = \left(\frac{1}{4}\right) * M_{i-1} \end{aligned} \quad (1)$$

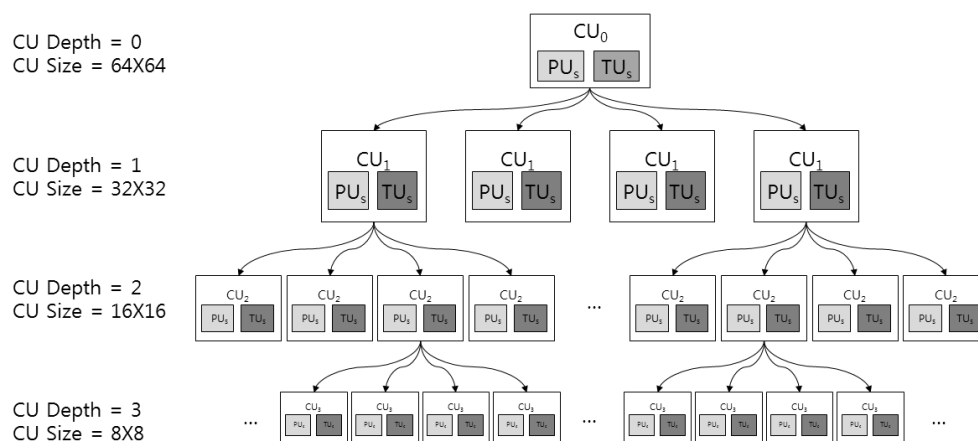


그림 1. CU 깊이와 크기에 따른 코딩 트리 구조에 대한 예

Fig. 1. Example of coding tree structure according to CU depth and size

여기에서, $f(n)$ 은 깊이 n 에 대해 요구되는 쿼드 트리 분할을 의미하며, M_i 는 i 번째 깊이에서 요구되는 CU의 쿼드 트리 분할을 의미한다. 식 (1)에서 알 수 있듯이, CU의 하위 깊이로 분할함에 따라 요구되는 계산 복잡도는 점진적으로 크게 증가함을 알 수 있으며, 각 CU의 크기에서 수행되는 예측 및 변환 과정을 고려해 볼 때, CU의 최대 깊이까지 분할을 수행하기 전에 최적의 CU 모드 또는 TU 및 PU에서의 RDO를 종료할 수 있다면, 부호화 속도 개선을 향상시킬 수 있다. 본 절에서는 HEVC 표준화 과정에서 제안되어 현재 HM에 선택적 옵션으로 채택된 Pruning 기반의 고속 부호화 알고리즘에 대해 소개한다.

그림 1은 앞서 설명한 HM에서 수행하는 최소 깊이 0부터 최대 깊이 3까지의 CU의 RDO 과정을 코딩 트리 형태로 나타낸 그림이다. [4]는 HM에서 수행하는 CU의 RDO에 대한 복잡도를 줄이고자, 화면 간 예측을 통한 부호화를 수행할 때, 64X64 CU0 대해 화면 간 예측 모드를 먼저 수행하여, MVD와 CBF (Coded block flag)가 모두 0일 경우 조기 종료하는 알고리즘을 제안하였다. [5]는 CU0이 SKIP 모드로 부호화 될 경우, 하위 깊이로의 분할을 수행하지 않는 알고리즘을 제안하였다. 다음 표 1은 HM에 선택적 옵션으로 채택된 Pruning 기반의 CU, PU, TU에서의 고속 부호화 방법이다^{[13][14]}. 이 외에도 고속 부호화 방법으로는 병렬화 기법을 통한 고속 HEVC 부호화 연구^[15], CU 구조에 대한 확률 기반 고속 부호화 방법^[16] 및 다양한 고속 부호화 방법^[17]들이 발표 되었다.

표 1. Pruning 기반의 HM 고속 부호화 모드
Table 1. List of fast encoding modes based pruning method

HM Parameter	Description
ESD ^[4]	Check inter 2NX2N PU first and terminate if motion vector difference (MVD) is zero and coded block flag equal to zero.
ECU ^[5]	No more CU split into the sub's CUs if skip mode is chosen
FDM ^[13]	Stop merge search if skip mode is used in Merge 2NX2N mode.
CFM ^[14]	Terminate encoding decision if partitioning PU has coded block flag equal to zero.

2. 전처리 기반의 고속 부호화 방법

[9]는 전처리 기반의 고속 부호화 방법을 제안하였다. 이는 부호화를 수행하기 이전에 원본 영상에 대한 화면 내 예측, 화면 간 예측을 수행하여, 부호화 과정에서 필요한 정보를 사전에 획득함으로써 부호화기의 속도를 향상과 더불어 율-제어 (Rate control)에서의 부호화 효율을 향상시키는 방법이다. [9]의 전처리 과정은 전처리 과정의 복잡도를 고려하여 원본 영상에 다운 샘플링 (Down sampling)을 수행하고, 이를 이용하여 화면 내 예측 및 화면 간 예측을 수행한다. 이때 화면 간 예측 수행을 통하여 획득한 움직임 정보를 기반으로 해당 블록이 부호화 과정에서 양방향 예측 또는 단방향 예측을 수행할지를 결정함으로써 부호화 속도를 향상시킨다. 또한, [9]는 전처리 과정을 통해 율-제어에서의 부호화 효율을 향상시키는 방법을 제안하며, 이는 화면 간 예측과 화면 내 예측을 전처리 과정에서 수행해봄으로써, 특정 블록이 실제 부호화기의 어떤 예측 모드로 부호화 될지를 예측한다. 예측된 결과를 기반으로 참조 블록으로써 높은 중요도를 갖는 블록은 많은 비트를 할당하고, 그렇지 않은 블록일 경우, 보다 적은 비트를 할당하여 율-제어에서의 부호화 효율을 향상시키는 방법을 제안하였다.

이 외에도 표 2와 같이 영상에 대한 필터링, 장면 전환 모듈 등을 전처리 과정을 수행함으로써, 부호화 과정에서 필요한 다양한 부호화 정보를 획득하여 부호화 속도 향상 및 효율을 향상시킬 수 있다. 본 논문에서는 부호화기의 속도 향상을 위해, 전처리 과정에서 움직임 추정을 통한 고속 부호화 방법을 연구를 진행하였다. 본 논문에서 제안하는 방법은 움직임 추정을 통해 움직임 벡터들을 계산하였고, 계층적 군집화 알고리즘을 적용, 각 CTU 마다 움직임 벡터의 군집도를 나타내는 군집화 비용을 계산하였다. 이를 통해 부호화 과정에서 각 CU의 최대 분할 깊이를 결정함으로써 CU의 RDO 과정의 복잡도를 줄이는 연구를 진행하였다. 제안하는 방법의 상세한 내용은 3장에서 소개한다.

표 2. 전처리 과정을 통해 기대할 수 있는 부호화 연구

Table 2. List of encoder study by using preprocessing

Preprocessing Module	Description	Effect
Scene change detection	Setting the reference list and structure depend on scene change	Coding efficiency and complexity reduction
Image enhancement filter	Applying the filter for image enhancement at coding picture	Coding efficiency
Estimation block weighting	Estimation of weighted factor for reference blocks by using the inter/intra prediction	Bit allocation for rate-control
Motion estimation	Decision of starting point on ME or MVP (derived ME process)	Complexity reduction
	Decision of prediction directions (Uni or Bi) on inter prediction mode	
	Prediction of CU split depth with MVs	
	Prediction of reference picture list	
Intra/Inter Prediction	Decision the prediction mode by using the inter/intra prediction result at the pre-processing	Complexity reduction

III. 제안하는 방법

본 논문에서는 전처리 과정에서 움직임 벡터에 대한 군집화를 통해 HEVC 부호화기의 부호화 속도를 향상시키는 방법을 제안한다. 제안하는 방법은 전처리 과정에서 부호화를 수행할 CU들에 대해 계층적 군집화를 수행하여, 군집화된 경향에 따라 최적의 깊이를 결정한다. 이후, 부호화 과정에서는 최적의 CU 깊이까지만 부호화를 수행하여 부호화기의 복잡도를 줄이는 방법이다. 본 장의 3.1에서는 본 논문에서 제안하는 전처리 과정을 통한 고속 부호화의 전체 알고리즘에 대해 설명하고, 3.2에서는 본 논문의 블록 깊이 결정을 위한 전처리 과정에 대해 설명한다. 마지막으로 3.3에서는 전처리 과정에서의 계층적 군집화 방법에 대해 설명하며 본 장을 마친다.

1. 움직임 벡터의 계층적 군집화를 이용한 고속 부호화 방법 소개

본 논문에서는 전처리 과정에서 움직임 벡터의 계층적 군집화를 이용하여 부호화 블록에 대한 최대 깊이를 결정한다. 이를 통해 부호화 과정에서 CU의 최적의 최대 깊이를 전처리를 통해 결정함으로써 부호화 속도를 향상시킨다. 제안하는 방법은 전처리 과정에서 움직임 벡터를 이용하여 계층적 군집화를 수행하며, 군집화된 경향에 따라 부호화

과정에서의 CU의 최대 깊이를 결정하였다. 현재 HEVC 참조 소프트웨어는 부호화 효율을 고려하여, 입력 영상을 CTU로 분할하여 순차적으로 분할한다. 이때, CTU는 쿼드 트리 형태의 최대 깊이의 CU로 계층적 쿼드 트리 분할하여 재귀적 부호화를 수행한다. 이러한 하향식 계층적 부호화 방식은 부호화 효율을 향상시킬 수 있지만, 부호화 과정에서의 큰 복잡도를 차지하는 원인 중 하나이다. 본 논문은 이러한 하향식 계층적 분할 방식을 해결하고자, 전처리 과정에서 움직임 벡터를 통한 계층적 군집화를 통하여 부호화 과정에서의 코딩 블록이 분할될 최대 깊이를 결정함으로써, 부호화 과정에서의 하향식 계층적 분할 과정에서 소비되는 복잡도를 줄여 부호화 속도를 향상시킨다. 그림 2는 본 논문에서 제안하는 전처리 과정에서 움직임 벡터 군집화를 이용한 고속 부호화 방법의 블록도를 나타낸다. 제안하는 방법은 크게 그림 2의 (a) 전처리 과정과 그림 2의 (b) 부호화 과정으로 구성될 수 있다. 전처리 과정에서는 원본 영상을 이용하여 다운 샘플링 (Down-sampling)을 수행, HEVC의 최소 블록 단위인 8x8 블록 단위로 움직임 추정을 수행하여 움직임 벡터를 계산한다. 이후, 본 논문에서 제안하는 방법인 움직임 벡터들을 지표로 삼는 계층적 군집화를 수행하여 실제 부호화를 수행할 블록의 최대 깊이 (Maximum CU Split depth)를 결정하는 과정을 수행하며, 보다 상세한 전처리 과정은 3.2와 3.3에서 설명한다. 부호

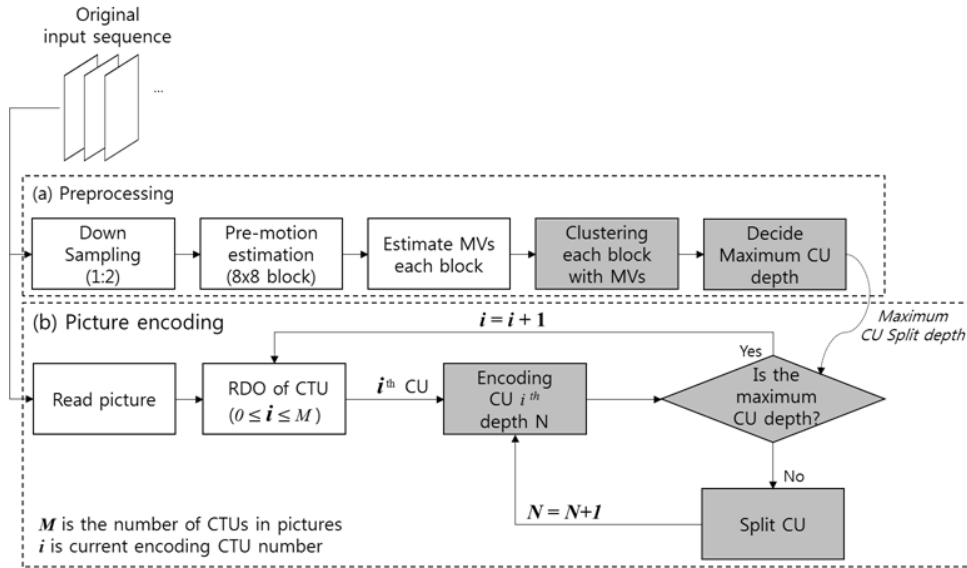


그림 2. 제안하는 전처리 과정을 이용한 고속 부호화 방법의 블록도

Fig. 2. Block diagram of the proposed method using preprocessing for the fast encoding

화 과정에서는 전처리 과정에서 전달받은 최대 분할 깊이 정보를 이용하여 CU의 최대 분할 깊이를 결정하여 부호화를 진행한다.

기존의 HM은 하나의 픽처 단위로 원본 영상을 입력받아 부호화를 진행하며, 하나의 픽처를 최대 CU 크기 (CTU의 크기)로 분할하고, CTU 단위로 부호화를 수행한다. 이때, 앞서 2장의 그림 1에서 설명한바와 같이 공통 실험 조건에서는 모든 CU에 대하여 최대 깊이까지 분할하여 각 CU에서 PU와 TU 단위로 예측과 변환 과정을 수행한다. 이러한 과정을 통해 픽처의 모든 CTU들에 대해 순서대로 하나의 CU에 대한 최적의 부호화 모드를 찾는 RDO 과정을 수행하며, 최종적으로 입력 영상에 대한 부호화를 마친다. 본 논문의 제안하는 방법을 적용한 부호화 과정에서는 전처리 과정에서 계산한 블록의 최대 분할 깊이를 각 CTU에 전달하여 해당 CTU에 대한 CU의 RDO과정에서 최대 분할 깊이를 적용한 부호화를 수행한다. 만약 최소 깊이를 0 (64x64 CU), 최대 깊이를 3 (8x8 CU)이라 가정할 때, i 번째 CTU에 해당하는 최대 분할 깊이가 2로 전달되면 해당 CTU는 64x64 CU로부터 16x16 CU 까지만을 분할하는 RDO를 수행하여 부호화를 수행한다.

2. 블록 깊이 결정을 위한 전처리 과정 소개

본 절에서는 부호화 과정에서 CU의 최대 깊이를 결정하기 위해 수행하는 전처리 과정을 단계별로 소개한다. 전처리 과정은 크게 5 단계로 구성되며, 부호화를 수행할 원본 영상을 입력으로 한다. 전처리 과정의 수행 결과는 CTU 단위로 최대 깊이를 결정하여 각 픽처별 부호화 단계에 적용된다.

그림 3은 본 논문에서 수행하는 전처리 과정의 블록도를 나타낸다. 전처리 과정은 (a) 다운 샘플링 단계, (b) 8x8 블록에 대한 움직임 추정 단계, (c) 블록의 움직임 벡터 계산 단계, (d) 각 블록에 대한 군집화 단계, (e) 최대 깊이 결정 단계로 구성된다. 먼저, (a) 다운 샘플링 과정은 부호화를 수행할 원본 영상을 1:2 비율로 해상도의 비율을 낮추는 과정이다. 그림 4는 다운 샘플링 단계를 수행했을 때, 코딩 블록의 비율을 나타낸 그림으로써, 그림 4의 (a)는 다운 샘플링을 수행했을 때, 원본 영상에서의 64x64 크기를 가지는 CTU에 대응하는 32x32 크기의 전처리 블록을 의미한다. 해당 전처리 블록은 16개의 8x8 크기를 갖는 최소 전처리 블록을 가지고 있다. 최소 전처리 블록 단위는 움직임

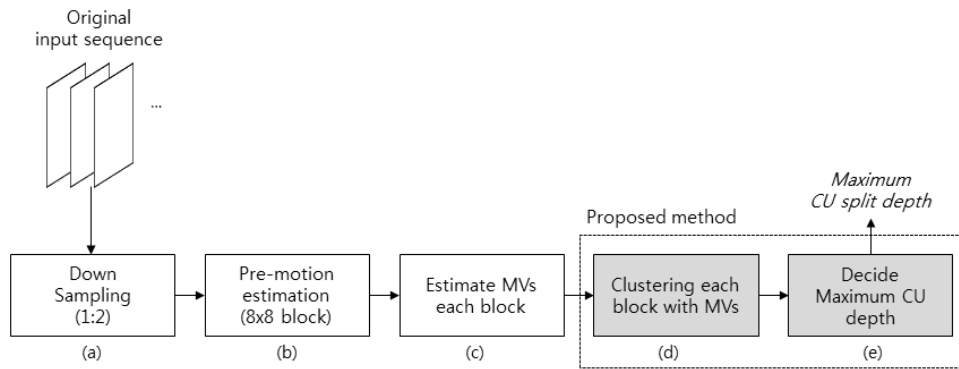


그림 3. 제안하는 전처리 과정의 블록도

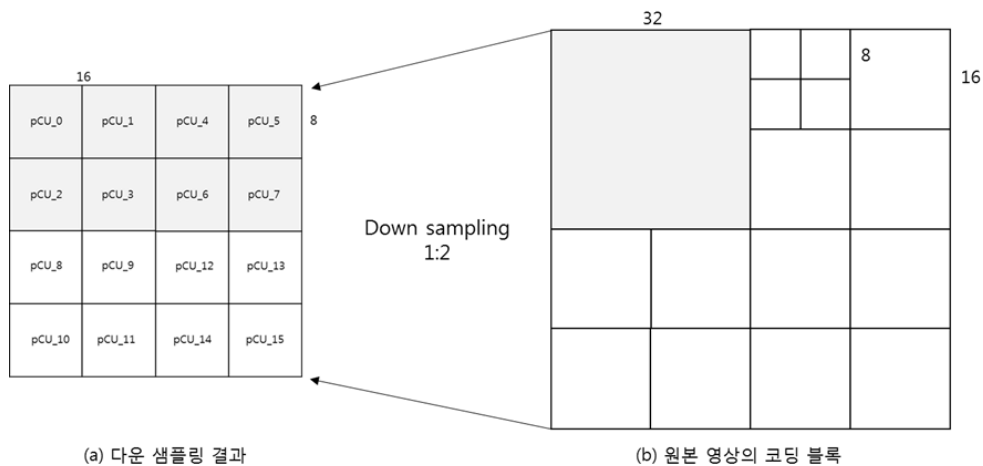
Fig. 3. Block diagram of the proposed preprocessing method

추정을 수행하는 단위이며, 16개의 최소 전처리 블록의 움직임 벡터를 군집화하여 군집화 비용을 계산한다. (b) 움직임 추정 단계에서는 이전 픽처의 탐색 영역 안에서 가장 작은 MAD (Mean Absolute Difference)를 찾는 전역 탐색 (Full search) 알고리즘을 이용하여 블록 정합 움직임 추정을 수행하였다. (b) 움직임 추정 단계에서는 이전 픽처의 탐색 영역 안에서 가장 작은 MAD를 찾는 전역 탐색 (Full search) 알고리즘을 이용하여 블록 정합 움직임 추정을 수행하였으며, 이전 프레임을 참조하여 식 (2)를 이용, 현재 블록과 가장 유사한 신호를 갖는 참조 블록을 찾아 움직임 벡터를 계산하였다.

$$M(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (f(m, n) - g(m + i, n + j))^2 \quad (2)$$

이때, M과 N은 움직임을 추정하는 블록의 크기를 의미하며, w를 탐색 범위라 할 때, i와 j는 $-w \leq i, j \leq w$ 의 크기를 가진다. 현재 블록과 가장 유사한 위치가 $i = x, j = y$ 인 경우, 현재 블록의 움직임 벡터 (MV)는 $MV(x, y)$ 가 된다. 이와 같은 방법을 통하여 (c) 움직임 벡터 계산 단계는 다운 샘플링한 영상의 모든 8x8 전처리 블록들에 대해 움직임 벡터들을 계산하는 과정을 수행한다.

움직임 추정 단계의 경우, 공간 해상도에서 이전 픽처 뿐



(a) 다운 샘플링 결과

(b) 원본 영상의 코딩 블록

그림 4. 원본 영상의 다운 샘플링 수행 결과

Fig. 4. Result of down sampling for original pictures

만 아니라, 실제 부호화 과정에서 참조하는 참조 구조를 고려하여 참조 픽처 리스트 (Reference picture list)를 구성, 이에 대한 움직임 추정을 수행할 수 있다. 또한, 움직임 추정 알고리즘에 있어, 본 논문에서 수행한 전역 탐색 이외의 고속 알고리즘을 적용할 수 있다. (d) 계층적 군집화 단계에서는 전처리 블록 단위의 계층적 군집화를 수행하여 16개의 최소 전처리 블록의 움직임 벡터들에 대한 평균 군집화 비용을 계산하게 된다. 그림 5는 계층적 군집화를 설명하기 위해 16개의 최소 전처리 블록 ($pCU_i, j = \{0, 1, 2, \dots, 15\}$)들의 움직임 벡터를 가정하여 그림으로 나타내었다.

그림 5와 같이 16개의 최소 전처리 블록은 영상의 특성에 따라 이전 참조 픽처에서 최소 MAD를 갖는 참조 블록을 나타내는 16개의 움직임 벡터를 계산하게 된다. 계층적 군집화 단계에서는 이러한 움직임 벡터를 갖는 최소 전처리 블록을 하나의 군집으로 설정하여, 움직임 벡터간의 유클리디언 거리 (Euclidean distance)가 유사한 군집 간에 병합하는 과정을 반복적으로 수행하여, 최종적으로 그림 6의 (a)와 같이 전처리 블록 당 하나의 군집 비용을 계산하게 된다. 그림 6의 (a)는 전처리 과정에서 하나의 전처리 블록에 대한 계층적 군집화 결과를 덴드로그램 (Dendrogram)으로 나타낸 결과이다. 그림 6의 (b)는 군집화 비용을 이용하여 부호화 블록에서의 최대 깊이를 설정하는 방법을 나타낸다. 최대 블록 깊이 결정 단계는 앞서 설명한 그림 6의 (b)와 같이 군집화 비용에 따라 대응하는 부호화 블록의 최

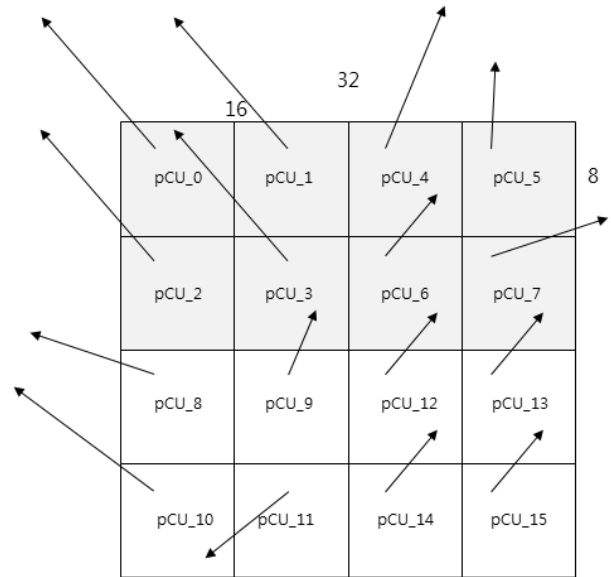


그림 5. 최소 전처리 블록의 움직임 벡터 예
Fig. 5. Example of motion vectors in the smallest preprocessing block

대 깊이를 설정하는 단계이다. 이러한 과정을 보다 상세하게 설명하면, 먼저, 8×8 크기를 가지는 16개의 최소 전처리 블록 그룹은 실제 부호화 과정에서의 CTU와 대응된다. 실제 계층적 군집화를 수행하기 위해서는 한 픽처 내의 모든 최소 전처리 블록들에 대한 움직임 벡터를 계산하며, 이후, 16개씩 그룹화 하여 군집화 결과를 산출하게 된다. 군집화된 결과를 산출하기 위해서는 하나의 최소 전처리 블록 내

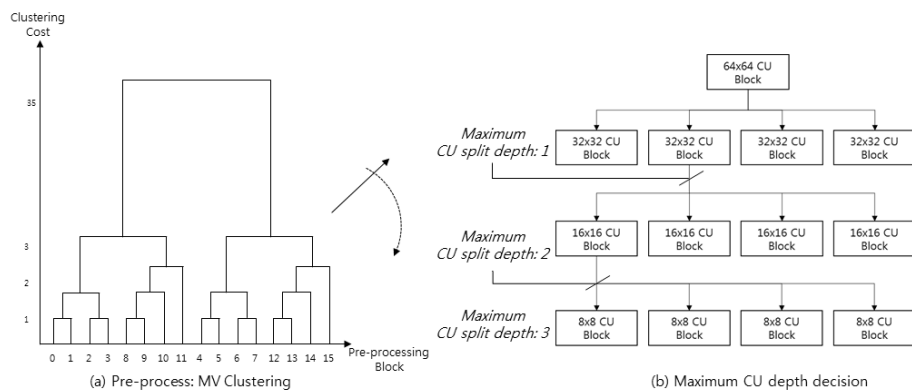


그림 6. (a) 계층적 군집화 수행 결과와 (b) 군집 비용을 이용한 최대 블록 깊이 결정
Fig. 6. (a) Result of Hierarchical clustering and (b) Maximum CU depth decision by using clustering cost

16개의 최소 전처리 블록들을 움직임 벡터(X, Y)를 대푯값으로 갖는 객체로 간주하며, 각 객체간 유클리디언 거리를 계산함으로써, 각 객체간의 군집 비용을 계산하게 된다. 이후, 군집 비용이 가장 가까운 객체를 하나씩 추가하는 방식으로 최소 전처리 블록 그룹에 대한 군집화 결과를 산출하게 된다. 이후, 최대 CU 깊이를 결정하는 과정은 계산된 군집 비용을 구간별로 나누어 일정한 구간별 임계치에 따라 최대 CU의 분할 깊이를 설정하였다.

이때, 본 논문에서는 군집화 비용에 따라 3단계의 최대 깊이를 설정하였다. 부호화 1단계 64x64 블록으로부터 32x32 블록까지의 계층적 부호화를 수행하며, 2단계에서는 64x64 블록으로부터 16x16, 마지막 3단계에서는 64x64 블록으로부터 8x8단위 블록까지 계층적 분할하여 부호화를 수행한다. 본 논문에서 이러한 전처리 과정을 통하여 부호화 과정에서 사용할 블록의 최대 깊이 정보를 획득하며, 부호화 과정에서는 전처리 과정에서 결정한 최대 블록 깊이를 이용하여 해당 깊이에 대한 RDO를 수행하여 고속 부호화를 수행한다.

3. 움직임 벡터를 이용한 계층적 군집화 방법 소개

본 논문의 전처리 과정에서는 움직임 벡터를 이용하여 계층적 군집화 방법 (Hierarchical clustering method)^[18]을

통해 최소 전처리 블록에 대한 군집화를 수행한다. 계층적 군집화 방법이란 주어진 데이터 셋을 계층적으로 병합하여 유사한 객체들의 군집으로 묶어 나가는 상향식 (Bottom-up) 군집화 방법으로써, 본 논문에서는 전처리 단계에서 최소 전처리 블록들을 하나의 움직임 벡터를 가지는 객체로 가정하여 계층적 군집화 방법을 적용, 전처리 블록 안에서의 움직임 벡터들의 경향을 파악하여, 블록의 분할 여부를 결정하였다. 본 논문의 계층적 군집화 과정에서는 군집과 군집 간의 거리 척도를 평가하기 위하여 평균 연결법 (Average linkage method)를 적용하였으며, 이는 식 (3)과 같다.

$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{u \in C_i, v \in C_j} d(u, v) \quad (3)$$

이때, C_i 는 군집화 과정에서의 i 번째 군집을 의미하며, $|C_i|$, $|C_j|$ 는 군집 i, j 의 개체수를 의미한다. $d(u, v)$ 는 객체 u 와 v 의 거리 또는 군집 간의 거리 척도를 의미하며, $D(C_i, C_j)$ 는 군집 C_i 와 군집 C_j 의 거리를 의미한다. 이러한 평균 연결은 군집 i 와 j 의 비유사성 척도로 두 군집간의 모든 객체 쌍의 평균 거리를 사용하기 때문에, 전처리 블록 안에서의 최소 전처리 블록들의 움직임 벡터의 군집화 정도를 평가할 수 있다. 그림 7은 전처리 블록에서의 최소 전처리 블록

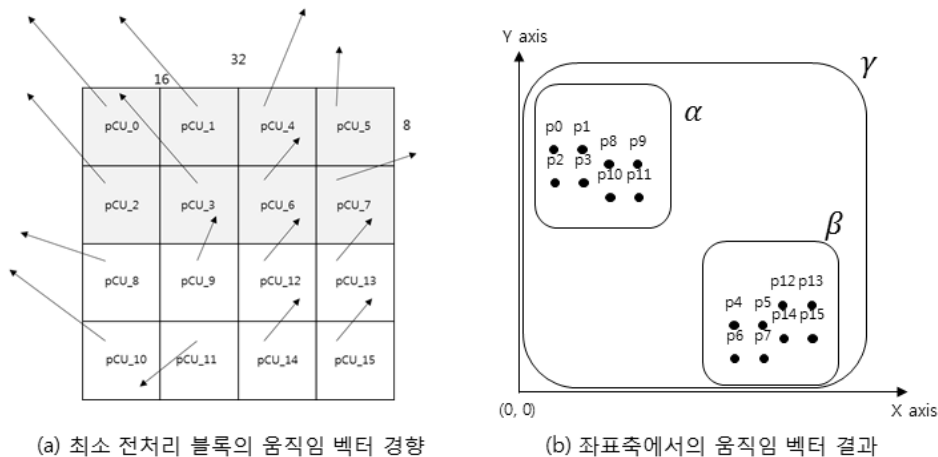


그림 7. (a) 최소 전처리 블록의 움직임 벡터 경향의 예, (b) (a)의 움직임 벡터를 X, Y 좌표축에서 표현한 예

Fig. 7. (a) Example of trend of motion vectors on the smallest preprocessing block, (b) showing of (a) on the coordinate X, Y axis

들이 움직임 벡터를 가질 때, X, Y 좌표 평면에서의 결과를 나타낸다. 그림 7의 (a)와 같이 최소 전처리 블록들이 크게 좌측 상단과 우측 상단으로 움직임 벡터가 나타날 경우, 이를 좌표축에서 나타내면, 그림 7의 (b)와 같이 나타낼 수 있다. 이때, 계층적 군집화를 수행하게 되면, 최종적으로 γ 에 해당하는 하나의 군집이 결정되고, 해당 군집 안에는 하위 군집에 속하는 α, β 가 군집화된다. 이때, 군집 γ 의 군집화 거리가 크게 나타날수록, 하위 군집들에 속한 개체들의 거리 차이가 크게 나타나며, 군집화 거리가 적게 나타날수록 하위 군집들에 속한 개체들의 거리 차이가 적다는 것을 의미할 수 있다.

본 논문에서 적용한 계층적 군집화 방법은 평균 연결법을 이용하여 개체간의 움직임 벡터의 유클리디언 거리가 가장 가까운 개체들을 하나씩 병합하여 군집하는 방법을 적용하였다. 이에 대한 결과는 앞서 설명한 그림 6의 (a)와 같이 군집화 된 결과를 덴드로그램 (Dendrogram)으로 나타낼 수 있다. 이는 이진 트리 (Binary tree)와 유사한 모형을 띄며, 개체 간의 거리가 가까운 개체들을 병합하고, 해당 병합 결과를 하나의 개체로 가정하여 반복적으로 다른 개체들과의 병합을 수행하는 계층적 군집화의 결과를 도식화한다. 본 논문에서는 이러한 계층적 군집화의 특성을 이용하여 군집화 비용을 전처리 블록 안에서의 움직임 벡터 경향으로 간주하였으며, 전처리 블록의 군집화 비용에 따라 CU의 최대 부호화 깊이를 설정하여 고속 부호화를 수행하였다.

IV. 실험 결과

본 논문에서 제안하는 전처리 과정을 통한 HEVC 부호화기의 고속화 방법의 성능을 평가하기 위하여 HEVC 참조 소프트웨어인 HM 16.0에서 공통 실험 조건 CTC (Common Test Condition)^[3]을 참고하여, 표 3의 실험 환경에서 제안하는 방법의 성능을 비교하였다. 성능 비교를 위해 진행한 실험은 HM 16.0과 본 논문이 제안하는 방법을 HM 16.0에 적용하여 평가하였다. 또한, 본 논문은 앞서 설명한 참조 소프트웨어에 적용된 고속 부호화 모드인 ESD

(Early Skip Decision)와 결합하였을 때의 제안하는 방법의 부호화 속도 향상을 비교하였다.

표 3. 실험 환경

Table 3. Experimental environments

CPU	Intel (R) Core (TM) i7 CPU 3930K 3.2 GHz
RAM	16.0 GB
OS	Windows 7 Ultimate K (64bit)

표 4는 본 논문의 실험 평가를 위해 사용된 실험 영상 정보를 나타낸다. Random Access Main Profile을 이용하여 Class B와 Class C 영상에 대한 실험을 진행하였으며, 표 4는 Class B의 실험 영상에 대한 세부 정보를 나타낸다.

표 4. 성능 평가를 위해 사용된 영상 정보

Table 4. Sequence information for performance evaluation

Class	Sequence name Resolution	Frame count	Frame rate	Bit depth	RA
B	Kimono 1920x1080	240	24fps	8	Main
	ParkScene 1920x1080	240	24fps	8	Main
	Cactus 1920x1080	500	50fps	8	Main
	BQTerrace 1920x1080	600	60fps	8	Main
	BasketballDrive 1920x1080	500	50fps	8	Main

표 5는 본 논문의 제안하는 방법인 전처리 과정을 통한 CU의 최대 분할 깊이 결정 방법을 HM 16.0에 적용한 부호화 결과와 HM 16.0을 비교한 결과이며, BD-rate[19]와 TS (Time Saving)로 성능을 비교하였다. 또한, 표 5에서의 TS은 제안하는 방법과 HM 16.0의 부호화 속도를 측정한 것으로 식 (4)와 같이 계산한다.

$$TS = \frac{T_{HM16.0} - T_{Proposed}}{T_{HM16.0}} \quad (4)$$

표 5. HM 16.0 대비 제안하는 방법에 대한 BD-rate와 부호화 속도 비교
Table 5. BD-rate and Time saving of the proposed method compared with HM 16.0

Class	Sequence	BD Rate(%)	Time Saving (%)
B	BasketballDrive	2.0	13.92
	BQTerrace	1.1	13.56
	Cactus	2.5	19.67
	Kimono	0.3	14.41
	ParkScene	1.4	18.38
Class B average		1.45	15.99
C	BasketballDrill	3.2	26.27
	BQMall	3.4	26.21
	PartyScene	3.4	27.79
	RaceHorses	2.8	26.11
Class C average		3.2	26.35

본 논문에서 제안하는 방법은 Class B에서 평균 1.45%의 BD-rate 손실이 있었으며, 평균 15.99%의 속도 향상이 있었다. 제안하는 방법은 영상의 특성에 따라 성능의 차이를 보인다. 가장 BD-rate 손실이 적은 “Kimono” 영상의 경우, 영상의 움직임 방향이 “BasketballDrive” 대비 일관성이 있으며, 영상의 배경과 물체 (Object)의 움직임이 천천히 움직이는 특성을 갖는다. 이러한 영상의 경우 전처리 과정에서 움직임 벡터를 지표로 하는 계층적 군집화를 통해 효과적으로 부호화 과정에서의 CU에 대한 RDO 복잡도를 줄일 수 있었다. 반면에, “BasketballDrive”와 같이 영상에서 물체가 빠르게 움직이는 경우, 다른 영상들 대비 더 많은 BD-rate 손실이 있었다. 이는 전처리 과정에서 수행한 움직임 추정을 수행할 때, 복잡도를 고려하여 실제 부호화 단계에서 사용되는 탐색 범위보다 제한된 탐색 범위를 설정하였기 때문이다. 이를 해결하여 부호화 효율을 향상시키기 위해서는 움직임 추정 단계에서의 탐색 범위 설정 및 움직임 추정 알고리즘 최적화, 해상도에 따른 최소 전처리 블록의 크기 조절 등을 통해 BD-rate 손실을 줄일 수 있다. 또한, 추가적으로 본 논문에서는 CTC에서 제공하는 Class C에 대하여, Class B에 적용한 최소 전처리 블록 크기를 적용하여 성능을 측정하였다. 4개의 실험 영상에서 평균 26%의 부호화 속도 감소가 있었지만, 평균 3.2%의 BD-rate 손실

이 있었다. 앞서 설명한 바와 같이, 해상도에 따른 최소 전처리 블록 크기는 본 제안하는 알고리즘의 부호화 효율과 부호화 속도를 조절하는 파라미터로 활용할 수 있으며, 이를 조절함으로써 목적에 따른 부호화를 수행할 수 있음을 확인하였다. 표 6은 CTC 조건에서 QP값을 22로 설정하였을 때, 1920x1080 (1080p)영상에서의 전처리 과정이 차지하는 시간 비율을 나타내며, 평균적으로 전체 부호화 시간에서 대비 전처리 과정이 차지하는 비율은 3.36%를 차지한다. 본 논문에서 제안하는 전처리 과정의 경우, 부호화 속도를 향상시키기 위하여 제한된 움직임 추정 알고리즘과 탐색 기법, 참조 픽처를 설정하여 전처리 과정이 차지하는 복잡도를 최소한으로 줄이고자 하였다.

표 6. 전처리 과정이 전체 부호화 과정에서 차지하는 비율 (QP 22)
Table 6. Ratio of preprocessing time over total encoding time (QP 22)

Resolution	Sequence	Preprocessing time ratio (%)
1920x1080	BasketballDrive	2.71
	BQTerrace	2.85
	Cactus	3.49
	Kimono	3.44
	ParkScene	3.66
Average		3.36

본 논문에서는 추가적으로, 기존의 HM에 선택적 고속 부호화 모드로 채택된 ESD (Early Skip Decision)[4]와 결합하였을 때의 성능을 비교하였다. 표 7은 이에 대한 실험 결과를 나타낸다. 먼저 표 7 (a)는 HM 16.0 대비 HM 16.0에 ESD 옵션을 적용하였을 때의 부호화 성능을 나타내며, 평균 0.3%의 BD-rate 손실과 31%의 부호화 속도 향상이 있음을 보였다. 표 7 (b)는 ESD 옵션을 적용한 HM 16.0과 본 논문에서 제안하는 방법과 ESD 옵션을 함께 적용하였을 때의 성능을 나타낸다. 이를 분석해보면, ESD 옵션만을 적용하여 부호화 했을 때 대비, 평균 1.54%의 BD-rate 손실이 있지만, 추가적으로 평균 16.95%의 부호화 속도를 향상시킬 수 있었다. 추가적인 속도 향상을 낼 수 있는 이유는 ESD 알고리즘에서 Pruning 되지 않는 CU 들에 대해 본 논문에서 제안한 CU의 최대 분할 깊이 값을 통해 Pruning 된

표 7. (a) HM 16.0 대비 ESD 알고리즘의 성능 비교 (BD-rate 및 부호화 속도), (b) ESD 알고리즘 대비 제안하는 알고리즘과 ESD를 결합한 방법의 성능 비교, (c) HM 16.0 대비 ESD 알고리즘과 제안하는 방법을 결합하였을 때의 성능 비교

Table 7. (a) BD-rate and time saving of the applied ESD on HM 16.0 compared with HM 16.0, (b) BD-rate and time saving of the combination method of proposed method and ESD method compared with applied ESD on HM 16.0, (c) BD-rate and time saving of the combination of proposed method and ESD method compared with HM 16.0

Class	Sequence	(a)		(b)		(c)	
		BD Rate(%)	Time Saving (%)	BD Rate(%)	Time Saving (%)	BD Rate(%)	Time Saving (%)
B	BasketballDrive	0.3	31.86	2.0	12.78	2.3	42.09
	BQTerrace	0.2	40.47	1.4	11.81	1.6	49.17
	Cactus	0.3	32.27	2.6	15.79	2.9	44.51
	Kimono	0.3	24.96	0.4	20.0	0.6	42.12
	ParkScene	0.2	28.03	1.5	24.36	1.7	47.74
Average		0.3	31.52	1.57	16.95	1.84	45.13

으로써, 추가적인 속도 향상을 낼 수 있었다. 마지막으로 표 7의 (c)는 HM 16.0 대비, 제안하는 기법과 ESD 알고리즘을 결합하였을 때의 성능을 비교하였다. 평균 1.84%의 BD-rate 손실이 있었으며, 45.13%의 부호화 속도 향상이 있었다. 본 논문에서는 부호화 속도를 위해 제한된 전처리 과정을 수행하였다. 부호화 효율을 향상시키면서 추가적으로 부호화 속도를 개선시키기 위해서는 앞서 설명한 바와 같이 전처리 과정의 움직임 탐색 범위와 움직임 탐색 알고리즘, 영상의 해상도에 따른 최소 전처리 블록의 크기 조절, 참조 픽처 구조 변경 등을 통하여 부호화기에서 수행하는 움직임 추정 단계와 보다 유사한 과정을 수행함으로써 부호화 효율을 향상시킬 수 있다. 또한, 전처리 과정과 부호화 과정을 병렬로 처리함으로써, 부호화 효율을 향상시키기 위해 수행할 때, 소요되는 추가적인 부호화 속도를 줄일 수 있을 것이다.

V. 결 론

본 논문에서는 움직임 벡터의 계층적 군집화를 통해 HEVC 부호화 고속 부호화 방법을 제안한다. 제안하는 방법은 전처리 과정에서 전처리 블록들에 대한 계층적 군집화를 수행함으로써, 부호화 과정에서 전처리 블록에 대응하는 부호화 블록의 최대 분할 깊이를 결정하여 부호화 속도를 향상시켰다. 본 논문에서 제안하는 방법은 1920x1080

영상에서 최대 19.67%의 부호화 속도를 개선함과 동시에 평균 1.45%의 BD-rate 손실을 보였으며, 기존의 고속 부호화 알고리즘과 결합하였을 때, 평균 1.84%의 BD-rate 손실을 보이면서, 45.13%의 부호화 속도 향상을 보였다. 본 논문에서 제안하는 전처리 과정에서 수행하는 입력 영상의 다른 샘플링 및 움직임 추정 과정을 GPU를 활용하여 병렬적으로 수행한다면, 전처리 과정에서 발생하는 추가적인 부호화 시간을 단축시킬 수 있다. 또한, 본 논문에서 제안하는 전처리 과정에 대한 연구를 확장하면 부호화 속도 개선 뿐만 아니라 Rate control을 위한 블록의 비트 할당 (bit allocation), 장면 변환 발생시, 참조 구조 개선, 화면 내 예측 과정을 통한 부호화 과정에서의 예측 모드 결정 등 다양한 관점에서의 HEVC 부호화 성능을 향상시킬 수 있는 연구를 수행할 수 있다.

참 고 문 헌 (References)

- [1] ITU-T Rec. H.165 and ISO/IEC 23008-2, "High efficiency video coding," Final draft approval Jan. 2013.
- [2] D. G. Sim, H. H. Jo, "Understanding of HEVC standard and technology," HongRung Publishing Company, Korea, 2015.
- [3] F. Bossen, "Common test conditions and software reference configurations", JCTVC-L1100, 12th JCT-VC meeting, Geneva, CH, Jan. 2013.
- [4] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, "Early SKIP Detection for HEVC," document JCTVC-G543, JCT-VC, Geneva, Switzerland, Nov. 2011.
- [5] K. Choi, S.-H. Park, and E. S. Jang, "Coding Tree Pruning Based CU

- Early Termination”, document JCTVC-F092, JCT-VC, Torino, Italy, Jul. 2011.
- [6] J.-S. Mok, Y.-J. Ahn, D.-G. Sim, “Adaptive fractional pel motion estimation for fast HEVC encoding,” 2015 Korean Society of Broadcast Engineers Summer Conference, pp. 268-270, June 2014.
- [7] J.-Y. Lim, Y.-J. Ahn, S. Park, D.-G. Sim, “Early termination algorithm in integer motion estimation for HEVC encoder,” 2014 IEIE Summer Conference, pp. 1849-1852, June 2014.
- [8] Yong-Jo Ahn, Tae-Jin Hwang, Dong-Gyu Sim and Woo-Jin Han, “Implementation of fast HEVC encoder based on SIMD and data-level parallelism,” EURASIP Journal on Image and Video Processing 2014, 2014:16, Mar. 2014.
- [9] Garrett-Glaser, J., “A novel macroblock-tree algorithm for high-performance optimization of dependent video coding in H.264/AVC,” available at http://x264.nl/developers/Dark-Shikari/MB-tree_paper.pdf
- [10] J. Lainema, F. Bossen, W.J. Han, J. Min, and K. Ugur, “Intra coding of the HEVC standard,” IEEE trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1792-1801, Dec. 2012.
- [11] P. Helle, S. Oudin, B. Bross, D. Marpe, M.O. Bici, K. Ugur, J. Jung, G. Clare, and T. Wiegand, “Block merging for quadtree-based partitioning in HEVC,” IEEE trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1720-1731, Dec. 2012.
- [12] Choi, Kiho, and Euee S. Jang. “Fast coding unit decision method based on coding tree pruning for high efficiency video coding.” Optical Engineering 51.3, 2012.
- [13] Frank Bossen, Benhamin Bross, Karsten Suhring, Davide Flynn, “HEVC Complexity and Implementation Analysis,” IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, vol. 22, no 12, Dec. 2012.
- [14] P. Onno and E. Francois, “Combination of J0171 and J0389 for the Nonnormative Encoder Selection of the Intra Transform Skip,” document JCTVC-J0572, JCT-VC, Stockholm, Sweden, Jul. 2012.
- [15] Yong-Jo Ahn, Tae-Jin Hwang, Dongkyu Lee, Sangmin Kim, Seoung-Jun Oh, Dong-Gyu Sim, “Study of Parallelization Methods for Software based Real-time HEVC Encoder Implementation” JBE Vol. 18, No. 6, pp. 835-549, November 2013.
- [16] Woo-Jin Han, “Conditional Probability Based Early Termination of Recursive Coding Unit Structures in HEVC” JBE Vol. 17, No. 2, pp. 354-362, March 2012.
- [17] ChanSeob Park, ByungGyu Kim, DongSan Jun, SoonHeung Jung, YounHee Kim, JinWook Seok, Jin Soo Choi, “Early Termination Algorithm of Merge Mode Search for Fast High Efficiency Video Coding (HEVC) Encoder” JBE Vol. 18, No. 5, pp. 691-701, November 2013.
- [18] Johnson, Stephen C. “Hierarchical clustering schemes,” Psychometrika, 32.3, 1967.
- [19] Bjøntegaard G (2001) VCEG-M33: calculation of average PSNR differences between RD curves, ITU-T SG16 Q6 Video Coding Experts Group (VCEG), Document VCEG-M33, Austin, Apr. 2001.

저 자 소 개



임 정 윤

- 2014년 2월 : 광운대학교 컴퓨터공학과 학사
- 2016년 2월 : 광운대학교 컴퓨터공학과 석사
- ORCID : <http://orcid.org/0000-0002-7035-8572>
- 주관심분야 : 영상압축, 컴퓨터비전, 영상신호처리



안 용 조

- 2010년 2월 : 광운대학교 컴퓨터공학과 학사
- 2012년 2월 : 광운대학교 컴퓨터공학과 석사
- 2012년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 박사과정
- ORCID : <http://orcid.org/0000-0002-0012-0905>
- 주관심분야 : 영상압축, 최적화 및 병렬화

저 자 소 개



심 동 규

- 1993년 2월 : 서강대학교 전자공학과 공학사
- 1995년 2월 : 서강대학교 전자공학과 공학석사
- 1999년 2월 : 서강대학교 전자공학과 공학박사
- 1999년 3월 ~ 2000년 8월 : 현대전자 선임연구원
- 2000년 9월 ~ 2002년 3월 : 바로비전 선임연구원
- 2002년 4월 ~ 2005년 2월 : University of Washington Senior research engineer
- 2005년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수
- ORCID : <http://orcid.org/0000-0002-2794-9932>
- 주관심분야 : 영상신호처리, 영상압축, 컴퓨터비전