적응적인 블록분할을 이용한 고해상도 영상의 고속 부호화

*이재영 **한종기 ***배진수

세종대학교

*ljy321456@naver.com, **hjk@sejong.ac.kr, ***baej@sejong.ac.kr

Fast Encoding Method for High Resolution Video using Adaptive Block Partition

*Lee, Jae-Yung **Han, Jong-Ki ***Bae, Jinsoo

Sejong University

요약

최근ISO/IEC와 ITU는 공동 협력팀(Joint Collaborative Team on Video Coding-JCT-VC)을 구성하여 HEVC(High Efficiency Video Coding)라 불리는 새로운 비디오 압축 표준 기술을 개발하고 있다. JCT-VC의 목표 중 하나는 H.264/AVC 압축률의 2배를 향상하는 것으로 최근 HEVC 테스트 모델(HEVC Test Model - HM)을 확정했다. HM의 여러 기술 중에서 확장 블록 구조 (large block structure) 기술은 CU(Coding Unit)와 TU(Transform Unit), PU(Partition Unit)로 구성된다. CU와 TU는 압축 단위와 변환 기술을 확장한 반복적인 문법구조(recursive syntax structure)이며, PU는 H.264/AVC과 동일한 형태를 띈다. 확장 블록 구조는 CU, PU, TU의 여러 조합에 의해 다양한 모드를 지원하여 압축 성능은 높아졌지만 HM 부호화기의 복잡도는 증가한다. 본 논문에서는 HM에 채택된 확장 블록 구조 기술에 대해 설명한 후, 계층적 B프레임 구조로 부호화 되는 경우 이전 레벨의 CU Depth 정보를 이용하여 현재 레벨의 CU Depth를 효과적으로 제한하여 기존의 방법보다 빠르게 부호화하는 방법을 제안한다.

1. 서론

MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)은 H.264/AVC의 표준화 이후, 고화질 및 고해상도 영상 압축을 위한 코덱의 필요성이 증가함에 따라 이를 만족시킬 수 있는 비디오 코덱의 표준화를 시작하였다. 그 첫 번째 과정으로, 2010년 1월 일본 Kyoto에서 열린 표준화 회의에서 JCT-VC (Joint Collaborative Team on Video Coding)을 설립하였다. 그리고 새로운 비디오 코덱 표준화 프로젝트인 HEVC (High Efficiency Video Coding)를 본격으로 진행하기 시작하였으며, 차세대 코덱 표준화를 위한 CfP (Call for Proposal)을 발표하였다[1]. 국내외 27개 기업, 연구소, 그리고 대학들은 CfP를 만족하는 비디오 코덱들을 설계 및 구현하여 제출하였으며, 2010년 4월 독일 Dresden에서 열린 첫 번째 JCT-VC 회의에 이 코덱들을 평가하고 논의하였다[3]. 그 후 2010년 10월 JCT-VC 중국 Guangzhou 회의에서는 참여기관들이 발표한 기술들을 기반으로 HM (HEVC Test Model)이 결정되었다[2].

MPEG-2를 비롯하여 MPEG-4, H.263 H.264/AVC 등의 기존 비디오 코덱들에서는 하나의 픽처(Picture)를 16×16 크기 블록 단위인 MB(Macro Block)으로 분할하여 부호화하였으나, HEVC에서는 부호화 기본단위인 CU (Coding Unit)을 사용한다. CU는 SPS(Sequence Parameter Set)에 저장된 파라미터값이 지정하는 최대 및 최소 크기까지 분할될 수 있다[5]. HM4.0에서는 최대와 최소 CU 크기를 각각 64×64와 8×8로 설정하여 부호화 및 복호화를 수행한다. 예측 부호화의 기본단위는 PU(Prediction Unit)로 정의되며, 하나의 CU는 다수개의 PU로 분할되어 예측된다. 예측 방법은 크게 인트라 (Intra) 예측과 인터 (Inter) 예측으로 분류할 수 있다. HM은 부호화 효율을 향상시키기 위하여 다양한 기술들을 사용한다. 인트라 예측에서는 최대 34가지 방향성을 고려한 인트라 예측과 평탄한 영역을 효율적으로 부호화하기 위한 Planar 예측이 사용되고 있으며, 인터 예측에서는 Merge 모드 및 AMVP(Advanced Motion Vector Prediction) 등의 기술들이 사용된다. 또한, 변환 부호화의 효율을 극대화하기 위하여 정수화된 DCT (Discrete Cosine Transform) 및 DST (Discrete Sine Transform)를 적응적으로 사용한다. 그 밖에 복원 영상의 화질 향상을 위하여 ALF(Adaptive Loop Filter) 및 SAO (Sample Adaptive Offset) 등의 기술들도 사용된다[5]. 이와 같이 JCT-VC에서는 다양한기술을 채택하여 HEVC 코덱의 부호화 효율을 향상시키기 위한 노력을 하고 있다. 이 과정에서, 부호화 효율뿐만 아니라 코덱의 복잡도도 동시에 고려하여 효율적인 Test Model을 정립하기 위한 다양한 실험과 분석을 수행하고 있다.

본 문서에서는 HM4.0의 계층적 부호화 구조를 구체적으로 서술하고, 제안된 방법과 HM4.0의 성능을 비교한다. 본 문서의 제 2장에서는 현재 HEVC에서 사용중인 계층적 부호화 구조를 설명하고, 제 3장

*본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [KI002142, 차세대 모바일 영상서비스를 위한 초경량 비디오 부호화 원천기술개발]

〈표 1〉 영상별 최적의 평균 CU Depth

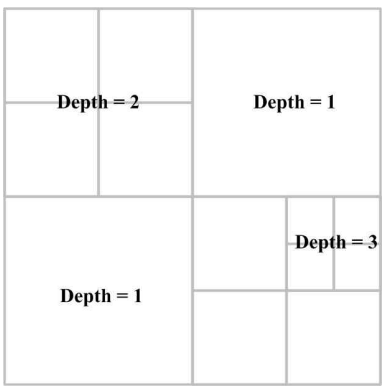
해상도	영상	AI-HE	AI-LC	RA-HE	RA-LC	LB-HE	LB-LC	LP-HE	LP-LC
Class A 2560x1600	Traffic	1.95	1.91	0.67	0.61				
	PeopleOnStreet	2.16	2.06	1.61	1.53				
	Nebuta	1.28	1.36	0.77	0.83				
	SteamLocomotive	0.83	0.82	0.34	0.33				
Class B 1920x1080	Kimono	1.15	1.07	0.65	0.60	0.82	0.79	0.84	0.81
	ParkScene	1.91	1.89	0.81	0.75	1.06	0.99	1.12	1.04
	Cactus	1.87	1.87	0.77	0.72	0.90	0.84	0.97	0.91
	BasketballDrive	1.66	1.74	0.72	0.71	0.84	0.86	0.89	0.91
	BQTerrace	2.08	2.08	0.76	0.73	0.90	0.87	1.10	1.06
Class C 832x480	BasketballDrill	2.32	2.35	0.97	0.94	1.04	1.02	1.13	1.09
	BQMall	2.23	2.28	0.95	0.91	1.15	1.12	1.23	1.19
	PartyScene	2.75	2.73	1.24	1.19	1.55	1.50	1.70	1.63
	RaceHorses	2.03	2.05	1.45	1.41	1.59	1.55	1.64	1.60
평균	Class A	1.55	1.54	0.85	0.83				
	Class B	1.74	1.73	0.74	0.70	0.90	0.87	0.98	0.95
	Class C	2.33	2.35	1.15	1.11	1.33	1.30	1.42	1.38

에서는 제안하는 방법에 대해 설명한다. 제 4장에서는 HEVC의 참조 소프트웨어인 HM4.0의 계층적 부호화 구조와 제안된 방법의 성능을 비교한다. 본 문서의 마지막 제 5장에서는 본 문서의 결론을 맺는다.

2. 계층적 부호화 구조 및 Hierarchical 프레임 구조

2.1 계층적 부호화 구조

기존의 코덱에서 기본적인 압축 기본 단위는 16x16 크기의 블록으로 사용하였으며, 이를 MB로 정의 하였다. HEVC에서는 기본 부호화 단위를 CU로 정의하며, CU의 최대 크기를 64x64로 확장하고 CU의 최소 크기를 8x8로 설정한다. <그림 1>와 같이 부호화기에서는 최대 크기의 CU부터 최소 크기의 CU까지 다양한 크기의 CU를 모두 사용하여 가장 효율이 좋은 크기의 CU를 사용하여 부호화를 수행한다. 이러한 CU의 화면간/화면내 예측 및 변환 모드는 입력 영상의 해상도와 복잡도, 움직임 정도와 같은 특성을 잘 반영하여 압축 성능을 향상한다.



〈그림 1〉 최적의 CU 크기 결정의 예

부호화 효율은 수식 (1)의 R-D (Rate-Distortion) Cost 최소화 관점에서 최적의 CU 크기를 결정한다.

$$J = D_Q + \lambda \times (R_H + R_C) \quad (1)$$

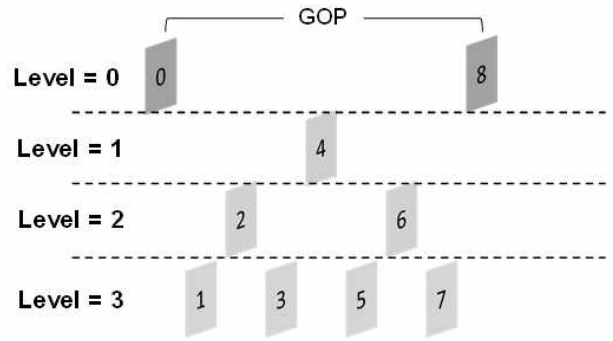
(1)에서 D_Q은 양자화로 인하여 발생하는 에러를 나타내 R_H은 예측 모드 및 움직임 정보 등을 부호화하기 위하여 사용되는 헤더 비트량을 의미한다. R_C은 양자화 된 변환 계수의 부호화에 사용되는 비트량을 나타낸다. λ은 라그랑지(Lagrange) 곱수이며, 양자화 에러와 사용된 비트량간의 상대적인 중요도를 표현하는 상수이다. 부호화기는 가장 작은값의 J를 발생시키는 CU 크기를 최적의 CU 크기로 결정한다.

HEVC에서는 다양한 크기의 CU를 사용함으로써, 영상의 공간 해상도 및 블록 특성을 효과적으로 고려하여 부호화할 수 있다. 일반적으로 영상의 해상도가 작거나 화소값들이 국지적으로 크게 변화는 경우에는 작은 크기의 CU들을 이용하여 인트라 및 인터 예측을 수행하는 것이 효율적이다. 작은 크기의 CU을 이용하게 되면 부호화에 필요한 헤더 비트량 R_H는 증가하지만, 상대적으로 예측이 정밀하게 이루어져 양자화 에러 D_Q와 변환 계수의 부호화에 필요한 비트량 R_C이 감소하는 장점이 있다. 반대로 영상의 공간해상도가 크거나 화소값들의 변화가 적은 영역에서는 큰 CU를 사용하는 것이 부호화 효율을 높일 수 있다. 이 경우에는 큰 CU를 사용하여도 작은 CU을 사용하여 예측하는 경우에 비하여 예측오차가 크게 증가하지 않는 경향이 있으므로 이러한 블록들을 부호화할 경우, 큰 CU를 사용하여 헤더 비트량 R_H을 절약하는 것이 효율적이다. <표 1>는 HM3.0을 이용하여 부호화한 영상별 최적의 CU Depth의 평균을 보여준다[6]. 값이 0이면 64×64 CU를 의미하고 3이면 8×8 CU를 나타낸다. <표 1>와 같이 영상의 해상도가 높으면 큰 CU가 많이 사용된다. 또한 한 영상의 특성에 따라 복잡한 영상들은 작은 CU로 부호화하는 것이 효율적이고, 움직임이 별로 없거나 복잡하지 않은 영상들에서는 큰 CU로 부호화하는 것이 효율적임을 알 수있다.

2.2 Hierarchical 프레임 구조

기존의 코덱에서 사용하던 Hierarchical B 또는 Hierarchical P 구조를 HM에서도 사용하고 있다.

보편적인 Hierarchical 구조는 <그림 2>와 같이 나타낼 수 있다. Hierarchical 구조의 부호화 순서는 낮은 레벨부터 부호화를 수행하고 가장 낮은 레벨의 있는 영상의 양자화 스텝을 기준으로 레벨이 증가함에 따라 양자화 스텝도 증가한다. 또한 영상을 부호화 할 때 자신이 속한 레벨보다 상위 레벨의 영상은 참조하지 못한다.



〈그림 2〉 Hierarchical B 또는 Hierarchical P 프레임 구조

3. 제안하는 방법

3.1 CU Depth의 분산을 이용한 블록의 분할 범위 결정

본 문서에서 제안하는 방법은 기존 HM에서 사용된 계층적 부호화 구조의 특징을 유지하면서 <그림 2>와 같이 Hierarchical B 구조 또는 Hierarchical P 구조일 때 이전 레벨의 CU Depth 정보를 이용하여 CU가 가장 작게 분할되는 Max_Depth를 효율적으로 제한하여 현재 레벨의 영상을 부호화 하는 방법이다.

먼저 하위 레벨에서 발생한 CU Depth 정보들을 수식 (2)을 이용하여 각 영상에서 발생한 평균 $Depth_{mean}(l)$를 구한다.

$$Depth_{mean}(l,k) = \frac{\sum_{i=0}^{N-1}(Block_{Area}(i) * Depth_i)}{Width * Height} \quad (2)$$

수식 (2)에서 l는 l번째 레벨을 뜻하고, k는 l번째 레벨에 존재하는 k번째 영상을 뜻한다. 또한 N은 영상 안에 있는 CU의 총 개수이고 $Block_{Area}(i)$는 i번째 CU의 넓이고, $Depth_i$는 i번째 CU의 Depth이다.

위와 같이 $Depth_{mean}(l,k)$을 구한 뒤에 하위 레벨에서 발생한 CU Depth의 분산 $V(l)$을 수식 (3)을 이용하여 구한다.

$$V(l,k) = \sqrt{\frac{\sum_{j=0}^{N-1}(Depth_{mean}(l,k) - Depth_j)^2}{N}} \quad (3)$$

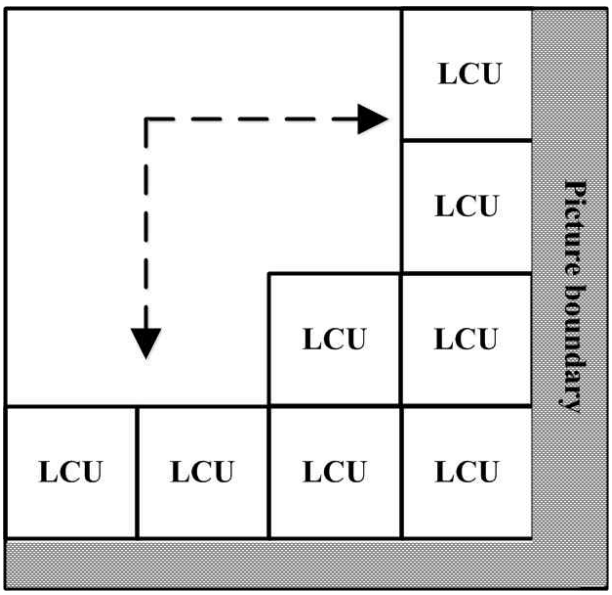
현재 레벨을 위한 Depth의 최대 범위 $R(l)$은 다음과 같이 구한다.

$$R(l,k) = \text{Round}(Depth_{mean}(l,k) + V(l,k)) \quad (4)$$

현재 영상의 CU Depth 범위를 결정하는 방법을 정리하면 다음과 같다.

(1) 최하위 레벨의 영상은 기존 HM과 같은 방법으로 부호화를 수행한다
(2) 차상위 레벨의 영상을 부호화하기 전에 가장 인접한 하위 레벨의 영상 중 같은 타입의 영상이 있는지 확인한다.
(3) 같은 타입의 영상이 존재한다면 하위 레벨 Depth 정보 $R(l,k)$를 현재 레벨의 최대 Depth범위로 설정한다.
(4) 같은 타입의 영상이 존재 하지 않다면 기존 HM의 방법으로 부호화를 한다.
(5) 최상위 레벨을 부호화 할 때까지 (2)에서 (4)를 반복한다.

3.2 영상의 경계에서의 블록의 분할 범위 결정



〈그림 3〉 영상 경계에서의 문제점

<그림 3>과 같이 보편적으로 영상 경계에서 LUC(Largest Coding Unit)는 영상 크기에 정확히 맞지 않는다. 그러므로 영상 경계 부근에서는 LCU를 남은 영상 크기에 맞게 분할해서 부호화를 수행하여야 한다. 그러기 위해서는 영상 경계 부근에서는 남은 영상의 크기와 현재 제안된 방법으로 제한하고 있는 CU Depth를 비교하여 다음과 같은 조건이 만족되면

$$(LCU_{Size} \gg \max Depth_{Current}) > Remain_{Size} \quad (5)$$

현재 레벨의 $\max Depth_{Current}$를 다음과 같이 재설정해야 한다.

$$Remain_{Size} = \log_2 2^{\alpha} \quad (6)$$

$$\max Depth_{Currnet} = \alpha$$

위와 같이 설정하면 영상의 경계에서도 적응적으로 CU의 Depth를 조절 하여 부호화 할 수 있다.

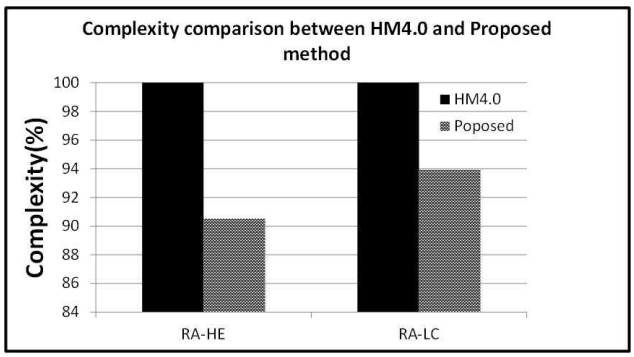
4. 실 험 결 과

실험은 Class C 영상 BasketballDrill, BQMall, PartyScene, RaceHorses을 하였다. 〈표 2〉에서 나타나는 것과 같이 하위 레벨의 CU depth 정보를 이용하여 현재 영상의 최대 CU Depth를 제한 하면 항상 최대 Depth로 부호화하는 기존의 방식과 비교 하였을 때 RA-HE와 RA-LC에서 평균 0.3%정도의 손실이 있었다.

〈표 2〉 BD-Rate comparison between HM4.0 and Proposed method

	RA-HE			RA-LC		
	Y	U	V	Y	U	V
BasketballDrill	0.4	0.6	0.8	0.4	0.5	0.8
BQMall	0.5	0.4	0.5	0.4	0.3	0.2
PartyScene	0.1	0.2	0.2	0.2	0.2	0.2
RaceHorses	0.1	0.3	0.3	0.1	0.2	0.1
Average	0.3	0.4	0.4	0.3	0.3	0.3

다음 <그림 4>와 같이 복잡도가 HM4.0 보다 감소했다. 이는 약간의 손실이 있어도 부호화기의 복잡도가 감소하였다.



〈그림 4〉 Encoder complexity complarison between HM4.0 and Proposed method

5. 결 론

Hierarchical 프레임 구조로 부호화가 수행되고 있을 때 이전 레벨의 CU depth정보로 다음 레벨의 depth를 제한하는 방법은 RA-HE와 RA-LC가 평균 0.3%의 손실이 있었고 복잡도는 약 6~10%의 감소가 있었다. 앞으로 슬라이스 단위가 아닌 블록단위로 적응적으로 CU Depth를 조절하는 것을 해볼수 있을 것이다.

참 고 문 헌

[1] "Joint Call for Proposals on Video Compression Technology," w11113, ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Picturesand Audio, Jan., 2010.

[2] "Summary of HEVC working draft 1 and HEVC test model (HM)," JCTVC-C405, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 3nd Meeting, Guangzhou, CN, 7-15 Oct., 2010.

[3] G. Sullivan and J. R. Ohm, "Meeting report of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC)," Dresden, DE, 15-23 April, 2010, JCTVC-A200, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting: Dresden, DE, 15-23 April, 2010.

[4] "Meeting report of the third meeting of the Joint Collaborative Team on Video Coding (JCT-VC),"JCTVC-C400, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 3rd Meeting, Guangzhou, CN, 7-15 Oct., 2010.

[5] "High Efficiency Video Coding (HEVC) text specification Working Draft 3," JCTVC-E603, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 5th Meeting, Geneva, CH, 7-15 Mar., 2011.

[6] C. W. Seo and J. K. Han "HEVC의 계층적 부호화 블록 구조," 電子工學會誌 第38卷 第8號, pp22~26, Aug, 2011