HEVC 부호화기 고속화를 위한 타일 기반 병렬화

김연희, 전동산, 정순흥, 석진욱, 최진수
한국전자통신연구원 방통융합미디어연구부 실감미디어연구팀
{kimyounhee, dschun, zeroone, jnwseok,jschoi}@etri.re.kr

Tile-based Parallelizing for a Fast HEVC Encoder

Younhee Kim, DongSan Jun, Soon-heung Jung, Jinwuk Seok, and Jin Soo Choi
Broadcasting & Telecommunications Convergence Media Research Department, ETRI

요 약

본 논문에서는 기존 AVC 보다 50% 압축성능 향상을 목표로 표준화가 진행되고 있는 차세대표준인 HEVC 부호화기의 속도를 높이기 위한 방안으로, HEVC 의 기술 중 화면 분할 기술인 타일(Tile)을 기반으로 효율적으로 부호화기를 병렬화하는 구조를 제안한다. 부호화기에서 복잡도가 높은 율왜곡 기반 모드 결정 과정을 멀티코어 병렬프로그래밍으로 구현하고, 병렬처리에 의한 속도 개선 결과를 제시한다. 타일은 병렬처리를 지원하기 위해 HEVC 가 채택한 구조로, 화면을 여러 개로 분할하여 부/복호화 할 수 있어 병렬처리 단위로 적합하며, 표준화의 기고서를 통해 화면분할로 인한 압축성능 변화량은 여러 차례 보고되고 있다. 본 논문의 결과에 의하면 타일의 수만큼 쓰레드를 생성하여 각 타일 단위로 율왜곡 기반 부호화 모드 결정을 하도록 병렬화 하였을 때 기존 참조 소프트웨어 대비 12 개의 쓰레드 생성 시 6 배의 속도 개선을 보인다. 향후 병렬로 처리할 수 있는 모듈을 확장하면 쓰레드 수 증가에 따른 속도개선 효과가 증대되어 부호화기 실용화를 위한 실시간 부호화기 개발에 한 걸음 다가갈 수 있을 것이라 기대한다.

1. 서론

영상 디스플레이 화면의 대형화와 더불어 고해상도 영상을 고효율로 압축하고자 하는 시장의 요구에 의해 JCT-VC(Joint Collaborative Team on Video Coding)은 기존 AVC/H.264 코덱보다 50% 압축성능 향상을 목표로 HEVC(High Efficiency Video Coding)로 명명한 차세대 코덱 표준화 작업을 2010 년에 시작하였다. 지난 2012 년 2 월에 열린 8 차 JCT-VC 회의에서 HEVC 의 압축효율 성능은 주관적 화질 기준으로 AVC 대비 평균 58% 압축성능 향상을 확인함으로써 목표수준에 달하였다고 발표하였다 [1].

HEVC 의 압축성능은 향상되었으나 코덱 특히 부호화기의 복잡도는 크게 증가하였다. 또한 4K 등의 고해상도 비디오 혹은 프레임 레이트(frame rate)를 높인 대용량 비디오를 HEVC 로 부호화하여, 방송 서비스나 VOD(Video on Demand) 서비스에 활용하기 위해서는 HEVC 부호화기의 고속화 방안 연구가 필수적이다.

현재의 많은 컴퓨터 프로세서는 2 개 또는 4 개 이상의 코어를 탑재한 멀티코어 기반으로 설계되기 때문에, 멀티코어를 활용한 병렬프로그래밍으로 HEVC 부/복호화기의 속도향상을 꾀할 수 있다. 이러한 하드웨어 발전 방향을 반영하여 HEVC 는 HW 및 SW 구현 시 병렬화를 쉽게 지원할 수 있는 기술들을 다수 채택하였다.

HEVC 가 채택한 병렬처리 지원 기술에 대한 연구는 주로 병렬처리로 인한 압축성능 영향 [2, 3, 4]과 디코더의 속도 개선 효과를 [5, 6] 중심으로 보고되고 있을 뿐 부호화기 관점에서의 연구는 아직 활발하지 않은 실정이다.

본 논문은 HEVC 의 기술 중 화면 분할 기술인 타일(Tile)을 기반으로 효율적으로 부호화기를 병렬화하는 구조를 제안하며, 그에 따른 속도 개선 결과를 소개하고자 한다.

2. HEVC 병렬화 지원 기술

HEVC 에 채택된 병렬화를 지원하는 기술 중 대표적인 기술로 엔트로피 슬라이스, WPP(Wavefront Parallel Processing), 타일(Tile)이 있으며, 본 장에서는 각각 기술에 대해 간단히 소개하겠다.

2.1 엔트로피 슬라이스

엔트로피 슬라이스는 엔트로피 코딩을 독립적으로 하는 단위이다. 즉 엔트로피 코딩은 엔트로피 슬라이스 단위로 하지만, 부호화 모드 결정이나 움직임 예측은 엔트로피 슬라이스 경계를 넘어 할 수 있는 단위이다.

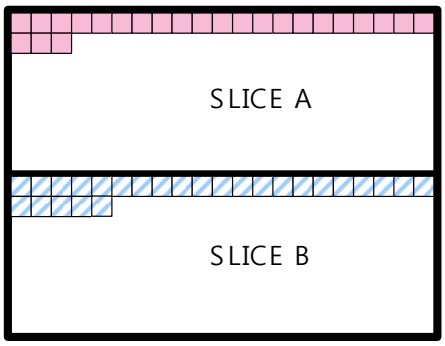

그림 1. 하나의 슬라이스 안에 두 개의 엔트로피 슬라이스가 포함된 예 [2]

엔트로피 슬라이스는 엔트로피 코딩 뿐만 아니라 부호화 예측도 독립적으로 하는 보통 슬라이스 단위로 병렬화 하였을 때보다 코딩 효율 감소폭을 줄일 수 있는 장점이 있지만, 경계부분의 부호화를 위해서는 지연(latency)를 감수하거나 버퍼링이 필요하다는 단점이 있다.

2.2 WPP

WPP 은 아래 그림 2 에서 보여주듯이, LCU(Largest Coding Unit) 행 별로 병렬화를 지원하는 구조이다. 코딩 효율의 감소폭을 줄이기 위하여 매 행 시작할 때, 상위 두번째 LCU 가 끝나는 시점의 CABAC 확률을 이용하여 엔트로피 코딩을 수행한다. 그러나 이러한 WPP 구조는 멀티코어를 기반으로 병렬화를 수행할 때 코어간의 통신을 필요로 하기 때문에 병렬화로 인한 오버헤드가 발생한다.

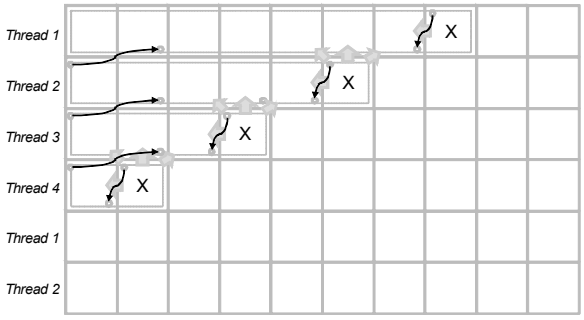

그림 2. WPP 에서 매 첫 행이 이용하는 CABAC 확률 위치 [3]

2.3 타일

타일은 아래 그림 3 에서 보여주듯이, 영상을 공간적으로 나누는 기법 중 하나로, 사각형 형태의 LCU 단위의 집합을 일컫는다. 타일은 LCU 수행 순서를 타일 내에서 래스터(Raster) 순서로 바꿔주기 때문에 메모리 제약적인 환경에서는 같은 조건의 슬라이스 기반의 화면 분할보다 더 넓은 움직임 탐색 범위를 허용할 수 있는 장점이 있다. HEVC 에서의 타일은 독립(independent)타일과 의존(dependent)타일 두가지로 설정할 수 있다. 독립타일에서는 슬라이스와 마찬가지로 부호화 예측과 엔트로피 코딩을 타일 단위로 독립적으로 수행하기 때문에 화면을 분할하여 멀티코어 기반 병렬화하기에 알맞은 구조이다. 반면 의존적인 타일은 부호화 예측 및 엔트로피 코딩이 모두 타일 경계를 넘어 이루어질 수 있기 때문에 압축성능 감소폭은 작으나 병렬처리의 어려움이 있다.

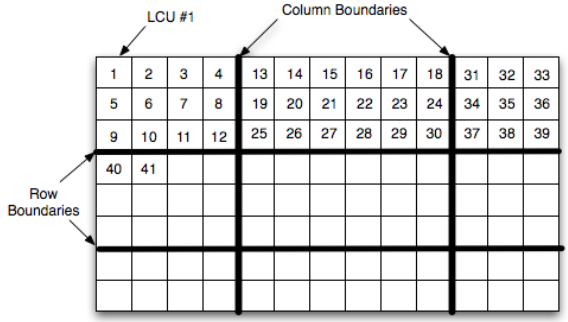

그림 3. 화면을 타일로 분할한 예 [4]

본 논문에서는 HEVC 참조 소프트웨어인 HM(HEVC Test Model) [7]에 독립타일 기반 멀티코어 병렬화를 적용하였다. 독립타일 단위의 부호화로 인한 압축 효율 결과는 [3]에 보고되고 있다.

3. 타일 기반 부호화기 멀티코어 병렬화

HM 에 구현된 부호화기는 LCU 단위로 부호화 모드를 결정하는 모듈과 실제로 부호화하는 모듈로 나뉜다. 전자의 부호화 모드를 결정하는 모듈에서는 CU(Coding Unit), PU(Prediction Unit), TU(Transform Unit) 별로 RD(Rate-Distortion) 관점에서 최적의 크기 및 부호화 모드를 결정하는 부분으로, 부호화기에서 가장 시간을 많이 소요하는 부분이기도 하다. 본 논문에서는 효율적인 부호화기 고속화를 위해 모드 결정하는 모듈을 타일 단위별로 OpenMP [1] 를 이용하여 각 코어에 쓰레드를 할당하여 병렬화 하였다.

현재 HM6.1 은 타일 구조를 적용했을 때의 코딩효율 측정을 목적으로 하고 있어, 단지 각 타일의 첫 LCU 에서 CABAC 의 확률을 초기화 할 뿐, 타일 순서대로 직렬 부호화하고 있다. 타일 단위로 병렬로 각 쓰레드에서 부호화 모드를 결정하도록 하기 위해서, 본 논문에서 제안하는 방법에서는 직렬로 부호화 할 때 사용하는 공유 데이터들(CABAC 관련 데이터 구조 및 QP 설정 구조)을 배열로 생성하여 각 타일 별로 할당하였다. 부호화 모드를 결정하는 모듈만 병렬화 하고 그 외 실제 모드 결정을 기반으로 부호화하는 모듈은 HM6.1 에서 구현된 방식을 그대로 적용하였다.

1	2	3	4	17	18	19	20
5	6	7	8	21	22	23	24
9	10	11	12	25	26	27	28
13	14	15	16	29	30	31	32
33	34	35	36	49	50	51	52
37	38	39	40	53	54	55	56
41	42	43	44	57	58	59	60
45	46	47	48	61	62	63	64

그림 5. HM6.1 에 구현된 타일기반 부호화 순서 및 CABAC 초기화

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

그림 6. 멀티코어를 이용한 타일 기반 부호화 순서 및 CABAC 초기화

[1] http://openmp.org/wp/

4. 실험결과

본 논문에서 제안하는 방법의 속도 향상 성능을 확인하고자 HM6.1 소프트웨어에 OpenMP 를 사용하여 부호화 모드를 결정하는 모듈을 병렬화하도록 구현하였다. 실험에 사용한 시스템의 사양은 표 1 과 같다.

표 1. 실험 시스템 사양

프로세서	Intel Core™ i7 CPU X 990 @3.47Ghz
코어 수	6 cores, 12 threads
메모리	24M RAM
Bus Speed	133.6 MHz
OS	64 bit Windows 7
컴파일러	Microsoft Visual Studio 2008

실험에 사용한 영상은 HEVC 표준에서 사용하는 실험 영상 중 Full-HD B 클래스 영상으로 각각의 영상 정보는 표 2 와 같다. 5 개의 영상 모두 해상도는 1920x1080 이며, 1 초에 해당하는 분량을 이용하여 HEVC 에서 정하는 메이(Main) 프로파일을 기준으로 표 3 과 같이 부호화 파라미터를 설정하고 속도를 측정하였다.

표 2. 실험 영상 정보

	실험영상 및 fps
B01	Kimono, 24 fps
B02	ParkScene, 24 fps
B03	Cactus, 50 fps
B04	BasketballDrive, 50 fps
B05	BQterrace, 60 fps

표 3. 실험에 사용된 부호화 파라미터 설정 값

부호화 파라미터	값
LCU 크기	64x64
CU 분할 깊이	4
GOP 크기	8
참조영상 수	4
엔트로피 코딩	CABAC
SAO	on
ALF	off
QP	22
TileInfoPresentFlag	1
UniformSpacingIdc	1
TileBoundaryIndependenceIdc	1
WaveFrontSynchro	0

쓰레드 개수에 따른 속도 개선의 효과를 측정하기 위해 쓰레드를 최대 12 개까지 생성하였고, 각 쓰레드의 개수마다 화면 분할하는 방법은 여러 가지 형태를 가질 수 있으나, 본 논문의 실험에서는 표 4 와 같이 화면을 분할하였다.

표 4. 쓰레드 개수에 따른 화면 분할 형태

Thread #	Tile # (행)	Tile # (열)
2	2	1
4	2	2
6	3	2
8	4	2
10	5	2
12	4	3

화면을 분할한 후, 각 영역에 쓰레드를 할당하여 부호화 모드 결정을 위한 모듈을 병렬처리 하도록 한 후 부호화에 걸리는 속도를 측정하였고, 각 실험 영상 별로 프레임 당 소요되는 부호화 시간을 초단위로 표 5 에 정리하였다. 부호화를 하는 과정 중에 모드를 결정하기 위한 RD 최적화 과정의 시간 복잡도가 높지만, 병렬로 그 모듈에 소요되는 시간은 감소하는 반면 직렬로 처리하는 부분을 변함이 없기 때문에 병렬처리로 인한 속도 감소의 폭은 줄어들고 있음을 그림 7 을 통해 알 수 있다.

표 5. 쓰레드 수와 프레임당 부호화 속도(sec.)

영상 / Thread 수	B01	B02	B03	B04	B05
1	53.03	44.23	48.73	54.81	54.16
2	27.60	23.18	26.56	30.24	27.95
4	14.09	12.29	13.87	17.37	15.77
6	10.85	8.97	10.69	13.32	11.89
8	9.51	8.27	9.73	11.25	10.69
10	8.96	7.70	8.83	10.65	9.59
12	8.77	7.41	8.56	9.77	9.17

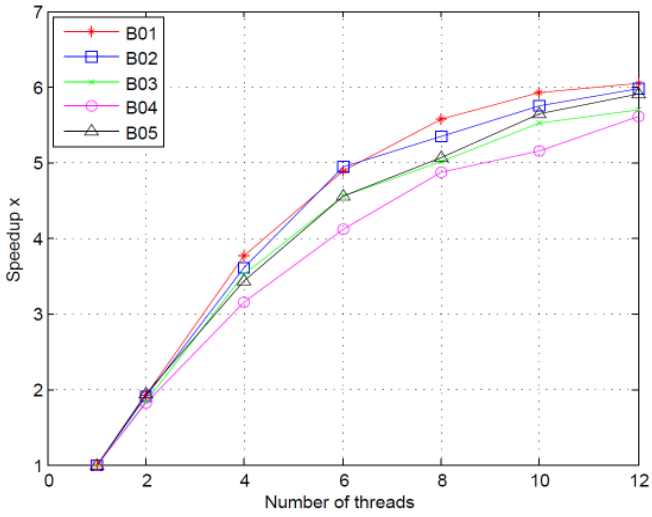

그림 7. 쓰레드 수 증가에 따른 부호화 속도 성능 향상

5. 결론

본 논문에서는 차세대 비디오 부호화 표준인 HEVC 부호화기의 고속화 방안으로 복잡도가 높은 RD 최적화 모듈을 타일 기반으로 병렬처리하도록 구현하고, 병렬처리로 인한 속도 개선 효과를 살펴보았다. HEVC 의 타일구조는 병렬처리를 지원하기 위하여 채택되었으나, 화면분할로 인한 압축성능 영향과 디코더의 속도 개선 효과를 중심으로 보고되고 있을 뿐 부호화기의 속도 개선 효과에 대해 정보가 부족한 실정이다.

율왜곡 최적화 모듈을 병렬처리로 구현한 본 논문의 실험 결과에 의하면, Full-HD 급의 해상도를 지닌 영상의 경우, 쓰레드 6 개를 생성하였을 때 최대 4.89 배의 속도 향상을 보였으며, 쓰레드 12 개를 생성하였을 때 최대 6 배의 속도 향상을 보였다. 향후 RD 최적화 모듈 이외에 병렬로 처리할 수 있는 모듈을 확장하면 쓰레드 수 증가에 따른 속도개선 효과가 증대될 수 있을 것으로 기대한다.

감사의 글

본 연구는 방송통신위원회의 ETRI 연구개발지원사업의 연구결과로 수행되었음. [KCA-2012-11921-02001]

참고 문헌

[1] Jens-Rainer Ohm, Gary Sullivan, Frank Bossen, Thomas Wiegand, Vittorio Baroncini, Mathias Wien, and Jizheng Xu, “JCT-VC AHG report: HM subjective quality investigation (AHG22),” JCTVC Document, JCTVC-H0022, San José, USA, Feb. 2012.

[2] Vivienne Sze, Madhukar Budagavi, "Analysis of entropy slices approaches," JCTVC Document, JCTVC-D243, Daegu, KR, Jan. 2011.

[3] Gordon Clare, Félix Henry, and Stéphane Pateux, “Wavefront Parallel Processing for HEVC Encoding and Decoding,” JCTVC Document, JCTVC-F274, Torino, IT, Jul. 2011.

[4] Michael Horowitz, Shilin Xu, Andrew Segall, and Minhua Zhou, “Tiles,” JCTVC Document, JCTVC-F335, Torino, IT, Jul. 2011.

[5] 유은경, 조현호, 서정한, 심동규, 김두현, 송준호, “HEVC 복호화의 LCU 기반 2D-Wavefront 병렬화,” IPIU, 2012, Feb.

[6] Mauricio Alvarez-Mesa, Chi Ching Chi, Ben Juurlink, Valeri George, and Thomas Schierl, “Parallel Video Decoding on the Emerging HEVC Standard,” ICASSP 2012, Mar.

[7] Ken McCann,Benjamin Bross, Il-Koo Kim, Kazuo Sugimoto, and Woo-Jin Han, “HM6: High Efficiency Video Coding (HEVC) Test Model 6 Encoder Description” JCTVC Document, JCTVC-H1002, San José, USA, Feb. 2012.