

OpenMP를 이용한 HEVC 디블록킹 필터의 병렬화 구현

*조현호 **서정환 ***유은경 ****심동규

광운대학교

*idjhh@kw.ac.kr

Parallel implementation of HEVC deblocking filter with OpenMP

*Hyun-Ho Jo **Junghan Seo ***Eun-Kyung Ryu ****Dong-Gyu Sim

Kwangwoon University

요약

본 논문에서는 OpenMP를 이용하여 HEVC 복호화기의 디블록킹 필터를 병렬화하는 것을 제안한다. 본 논문에서는 HEVC 디블록킹 필터를 병렬화하기 위하여 슬라이스를 병렬 처리가 가능한 코어의 개수만큼의 영역으로 균등하게 분할 한 후 각 영역에 코어를 할당하였다. 각 영역에 할당된 코어들은 자신의 영역 내의 LCU에 대해서 순차 주사 순으로 필터링을 수행하는데, 먼저 영역 내의 모든 LCU에 대하여 수평방향으로 필터링을 수행한다. 이러한 수평방향 필터링이 완료된 후 동일한 영역에 대하여 다시 수직 방향으로 필터링을 수행한다. 본 논문에서 제안하는 OpenMP를 이용한 HEVC 디블록킹 필터 병렬화를 통하여 4-Core 환경에서 복호화기에서 디블록킹 필터링의 수행 시간을 약 2.5배 감소 시켰다.

1. 서론

HEVC(High Efficiency Video Coding)는 JCT-VC(Joint Collaborative Team on Video Coding)에 의해 표준화가 진행 중인 차세대 비디오 코딩 기술이다.[1] HEVC는 H.264/AVC High 프로파일 대비 약 50%의 비트 감소를 목표로 표준화가 시작되었으며 현재 부호화 효율 향상을 위한 다수의 기술들이 사용되고 있다. 먼저, HEVC는 MPEG-2, MPEG-4 그리고 H.264/AVC와 같은 기존의 비디오 코덱들과 달리 매크로블록을 사용하지 않는다. 그 대신 HEVC는 CU(Coding Unit), PU(Prediction Unit), TU(Transform Unit)를 정의하고, 이를 통하여 다양한 입력 크기의 영상을 효율적으로 부호화한다. 인트라 예측에서는 4×4, 8×8, 16×16, 32×32, 64×64과 같은 PU 크기에 따라 서로 다른 예측 모드의 개수를 사용한다. 예를 들어, 32×32 PU에서는 최대 33개의 방향성을 갖는데 이는 H.264/AVC의 인트라 예측보다 더 세밀하게 방향을 예측하여 부호화 성능을 향상 시킨다. 인트라 예측에서는 PMV(Predicted Motion Vector)를 명시적으로 시그널링하는 기술, 모션 벡터와 참조 프레임 인덱스를 주변에 위치하는 블록과의 병합(Merge)하는 기술 등을 사용한다. 또한, HEVC에는 H.264/AVC의 후처리 필터링 기술인 디블록킹 필터에 추가적으로 SAO(Sample Adaptive Offset), ALF(Adaptive Loop Filter)라는 기술을 사용함으로써 부호화 효율 및 주관적 화질을 향상 시킨다.

HEVC는 앞에서 설명한 다양한 기술들을 사용함으로써 H.264/AVC 하이 프로파일 대비 약 43%의 비트 감소를 보이지만, 움직임 보상 과정에서의 8-tap 필터, 디블록킹 필터, SAO, ALF의 세 가지의 후처리 필터의 사용으로 인해 복호화기의 계산량을 증가 시킨다. HM 3.0을 기준으로 HEVC 복호화기의 복잡도를 분석한 결과 HE(High Efficiency) 조건에서 약 40%의 계산량이 디블록킹 필터,

SAO, ALF에서 발생하였다. 따라서 실시간 HEVC 복호화기를 구현하기 위해서는 먼저 이러한 세 가지의 후처리 필터링 기술에 대하여 고숙화하는 것이 필수적이다. 본 논문에서는 세 가지의 후처리 필터링 기술 중 디블록킹 필터링에 대하여 OpenMP를 사용하여 LCU(Largest Coding Unit) 단위의 데이터 레벨 병렬화를 적용함으로써 복호화 속도를 향상 시켰다.

본 논문의 구성은 다음과 같다. 2장에서는 HEVC의 디블록킹 필터의 특징과 OpenMP를 이용한 병렬화 구현을 소개한다. 3장에서는 구현된 병렬 디블록킹 필터의 성능을 평가 및 분석하고, 4장에서는 본 논문에 대한 결론 및 향후 연구의 진행 방향을 제시하고 본 논문을 마치도록 한다.

2. HEVC 디블록킹 필터 병렬화

가. HEVC 디블록킹 필터링

HEVC의 디블록킹 필터는 H.264/AVC의 디블록킹 필터와 유사한 구조를 가지는 인-루프 형태의 적응적 필터이다. HEVC의 디블록킹 필터는 필터링을 수행할 블록 경계에서 주변 블록의 부호화 정보 및 부호화된 픽셀 값을 사용하여 필터링의 적용 여부, 필터링의 강도 및 필터의 종류를 결정한 후 필터링을 수행한다. HEVC의 디블록킹 필터는 인-루프 형태의 필터이므로 복호화기에서도 부호화기와 동일하게 이러한 과정을 수행한다. 디블록킹 필터는 이러한 적응적 필터링을 위하여 블록의 경계에서 Bs(Boundary Strength)값을 계산하는데, 이때 블록 경계를 기준으로 인접하는 블록의 타입, DCT 계수의 포함 여부, 모션 벡터 및 참조 프레임 인덱스 등의 값을 참조한다. 이러한 Bs 결정 과정은 다수의 분기문을 사용하기 때문에 복잡도가 높다. 또한, 필터링을 수행하는 과정에서도 메모리에서 픽셀

값을 읽어온 후 필터링을 수행하고, 그 결과 값을 메모리에 다시 쓰기 때문에 높은 복잡도를 갖는다. 이러한 디블록킹 필터의 높은 연산량을 고려하여 HEVC의 디블록킹 필터는 필터링의 연산 복잡도를 낮추기 위하여 8x8 블록 경계 이상에 대해서만 필터링을 수행한다.

나. OpenMP를 이용한 병렬화

H.264/AVC의 디블록킹 필터는 매크로블록 단위에서 수평, 수직 방향으로 필터링을 수행한다. 따라서 인접하는 매크로블록 간의 의존성이 존재하며, 이는 H.264/AVC의 디블록킹을 매크로블록 단위의 데이터 병렬화를 할 때 2D-Wave 방식을 사용하여 병렬화하게 한다.[2] 그러나 이러한 2D-Wave 방식은 슬라이스 단위에서 병렬화의 시작 부분과 끝부분에서 동시에 수행 가능한 매크로블록의 수를 감소시키기 때문에 병렬화의 성능이 최대가 되지 않는다. 이와 달리 HEVC의 디블록킹 필터는 표준화 단계에서부터 디블록킹 필터링을 고속화하기 위하여 디블록킹 필터의 병렬처리를 고려하였다. 이에 따라 HM 3.0 이후부터는 디블록킹 필터링을 수행할 때 슬라이스 내의 모든 필터링 경계에서 수평 방향으로 디블록킹 필터링을 수행한 후, 다시 수직 방향으로 디블록킹 필터링을 수행한다. 또한, 필터링 경계에서 최대 세 픽셀씩 필터링을 하기 때문에 H.264/AVC와 달리 인접하는 필터링 경계에서 의존성이 존재하지 않는다. 따라서 HEVC는 2D-Wave 방식을 사용하지 않아도 병렬처리가 가능하며, 주변 블록들과 의존성이 없기 때문에 2D-Wave 보다 더 높은 병렬화 성능을 갖는다.

본 논문에서는 그림 1과 같이 슬라이스를 코어의 개수만큼의 영역으로 나눈 후 각 영역에 코어를 할당하였다. 각 영역에 할당된 코어는 자신의 영역 내에 있는 LCU 들에 대하여 디블록킹 필터링을 수행한다. 하나의 코어가 자신의 영역 내의 LCU들에 대하여 디블록킹 필터링을 수행하는 경우, 의존성이 존재하지 않기 때문에 영역내의 LCU들에 대하여 임의의 순서대로 처리할 수 있다. 그러나 그림 1과 같이 순차 주사 순으로 LCU를 처리하는 것이 메모리 액세스를 고려할 때 임의의 순서보다 더 효과적이다. 본 논문에서는 그림 1과 같이 코어들을 각 영역에 할당한 후 먼저 수평 방향으로 디블록킹 필터링을 수행하였다. 각 영역에 할당된 코어는 자신의 영역 내에 있는 모든 LCU에 대하여 수평 방향으로 필터링을 수행한 후 슬라이스 내의 다른 영역에 대한 작업이 끝날 때 까지 기다린다. 슬라이스 내의 모든 영역에서 수평방향으로 필터링이 완료 된 후 동일한 영역에 대하여 수직 방향으로 필터링을 수행한다. 수직 방향으로 필터링을 끝낸 코어들은 슬라이스 내의 모든 영역에서 수직 방향으로 필터링이 완료될 때 까지 대기한 후, 또 다른 후처리 필터인 SAO, ALF를 수행한다.

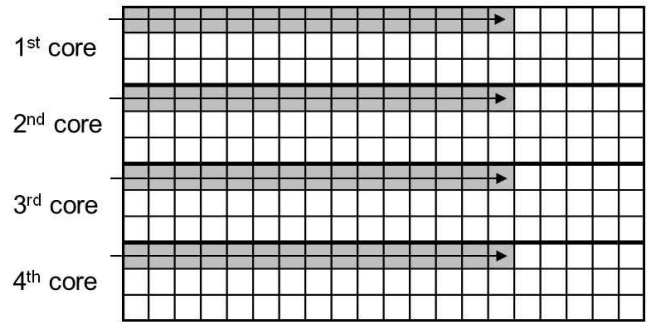


그림 1. LCU 단위의 디블록킹 필터 병렬화

3. 실험 결과 및 분석

본 논문에서 제안하는 HEVC 디블록킹 필터 병렬화의 성능을 확인하기 위하여 HM 3.0 참조 소프트웨어를 OpenMP를 사용하여 병렬화 하였다. 실험 환경은 마이크로소프트 윈도우 7(64 비트)기반에서 Intel Quad Core 프로세서를 사용하였다. 실험 영상으로는 Full-HD(1920x1080) 해상도를 갖는 B class의 BasketballDrive, BQTerrace, Cactus를 사용하였으며 각 영상이 12Mbps가 되도록 비트레이트를 조절하였다.

제안하는 디블록킹 필터의 병렬화의 실험 결과는 표 1과 같다. 병렬화에 따른 성능을 확인하기 위하여 HM 3.0 소프트웨어를 싱글 코어에서 동작 시켰을 때의 디블록킹 필터 수행 시간과, 이를 다시 4개의 코어를 사용하여 병렬처리를 통해 동작 시켰을 때의 디블록킹 필터 수행 시간을 비교하였다. HEVC의 디블록킹 필터에 대하여 수평 방향 필터링과 수직 방향 필터링 각각에 대하여 LCU 단위의 순차 주사 순으로 필터링을 적용하는 병렬화를 수행한 결과 4개의 코어를 사용하였을 때 평균적으로 2.51배의 디블록킹 필터링 수행 시간이 감소됨을 확인할 수 있다.

표 1. 병렬 디블록킹 필터링 수행 시간

시퀀스	디블록킹 필터 수행 시간 (sec)		수행시간 감소 비율
	1-Core	4-Core	
Basketball Drive	0.666	0.263	2.53
BQTerrace	0.649	0.249	2.61
Cactus	0.683	0.284	2.40
Average	0.666	0.265	2.51

4. 결론

본 논문에서는 HEVC 복호화기에서 높은 연산량을 갖는 디블록킹 필터에 대하여 병렬화를 통해 수행 시간을 감소 시켰다. HEVC의 디블록킹 필터의 병렬화를 위하여 슬라이스를 코어의 개수만큼의 영역으로 균등하게 분할 후, 각 영역에 코어를 할당하여 병렬처리 하였다. 또한, 영역 내에서 LCU 단위로 디블록킹 필터링을 수행할 때 순차 주사 순으로 필터링을 수행하도록 하였다. 이렇게 디블록킹 필터링 연산을 4개의 코어를 사용하여 병렬 처리함으로써 디블록킹 필터링 과정의 수행 시간을 약 2.51배의 감소 시켰다. 추후에는 SIMD(Single Instruction Multiple Data)와 같은 명령어

레벨 최적화와 추가적인 알고리즘 레벨 최적화를 동시에 적용하는 고속 디블록킹 필터링에 대하여 연구할 것이다.

감사의 글

"본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음"
(NIPA-2011-C1090-1100-0010)

참고문헌

- [1] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp.1661-1668, Dec. 2010.
- [2] C. Meenderinck, A. Azevedo, M. Alvarez, B. Juurlink, and A. Ramirez, "Parallel Scalability of H.264," *Proceedings of the first Workshop on Programmability Issues for Multi-Core Computers*, January 2008.