

비대칭 코어를 위한 HEVC 병렬처리에서의 복잡도 예측 기반 타일 할당

노현준, 류영일, 김선대, 이상순, 류은석

가천대학교 컴퓨터공학과

ggyo@gc.gachon.ac.kr, wlrmlrm99@gc.gachon.ac.kr, ele7004@gc.gachon.ac.kr,
sslee@gachon.ac.kr, esryu@gachon.ac.kr

Prediction Complexity-Based Tile Allocation for Asymmetric Multicores in HEVC Parallel Processing

Hyun-Joon Roh, Yeongil Ryu, Seon-Dae Kim, Sang-Soon Lee, Eun-Seok Ryu

Department of Computer Engineering, Gachon University

요약

본 연구는 비대칭 모바일 코어(Asymmetric mobile cores)의 연산 능력을 고려한 새로운 병렬 영상 처리 방법을 제안한다. HEVC(High Efficiency Video Coding) 예측 유닛인 PU(Prediction Unit)의 분할 정도를 기반으로 각 타일(Tile) 별 처리 복잡도를 예측하고, 이 예측된 복잡도가 비대칭 코어들의 연산 능력에 따라 최적화 되도록 할당한다. 구현 및 실험을 통한 검증을 통해 PU의 분할 정도와 연산 복잡도 (디코딩 시간) 사이에 높은 상관관계가 있다는 것을 확인하였고, 12개와 18개 타일로 분할된 HEVC 영상을 6개의 각 비대칭 모바일 코어에 적응적으로 할당하여 디코딩 한 결과 최대 약 17%의 처리속도 향상을 확인하였다.

1. 서론

최근 UHD(Ultra High Definition) 영상 콘텐츠들의 보급이 활발히 진행되고 있는데, 2016년에 Cisco가 발표한 VNI(Visual Network Index) Global IP Traffic Forecast, 2015-2020라는 자료에 따르면 2020년에는 전세계 IP 비디오 중 UHD 영상이 점유하는 비율이 2015년의 점유 비율 2.0%의 약 8배나 되는 16.4%의 사용량을 보일 것으로 예상된다[1]. 이러한 UHD영상은 이전에 많이 사용된 FHD(Full High Definition)의 4배 크기(8K UHD는 16배)이기 때문에 이를 지연 없이 실시간으로 처리하기 위해서는 더욱 높은 성능이나 더 우수한 메모리가 필요하다.

우리는 UHD영상의 영상처리 속도를 개선하기 위해 새로운 비대칭 코어에서의 타일 기반 병렬처리 방식을 제안한다. 타일이란 2013년 1월에 국제표준화 단체인 ISO/IEC와 ITU-T의 공동 구성체인 JCT-VC에서 표준화한 HEVC의 병렬처리 기술 중 하나로 한 화면을 여러 직사각형들로 분할하여 처리하는 기술이다[2].

비대칭 코어는 한 모바일 CPU에 존재하는 멀티코어들 중 일부는 성능이 더 좋은 코어들(이하 빅코어), 일부는 성능이 덜 좋은 코어들(이하 리틀코어)로 구성된 것을 의미한다. 최근 ARM에서 출시하는 CPU 중 big.LITTLE 아키텍처로 구성된 것이 비대칭 멀티코어 CPU이다[3]. 비대칭 코어는 처리할 작업의 연산량에 더 적합한 코어를 사용함으로써 일반 대칭코어보다 전력소모를 줄이는 장점이 있다. 하지만 비대칭 코어로 병렬 영상처리를 수행하는 현재 기술들은 그 빅/리틀 구조의 장점을 살리지 못하고 있다. 현재 사용중인 멀티코어에서의 병렬 영상처리 시스템은 각 코어의 성능비를 고려하지 않고 순차적으로 타일을 할당한다. 그 결과 각 코어에 할당된 작업량과 코어의 성능비가 일치하지 않기

때문에 효율성이 매우 떨어진다. 예를 들어, 각 코어에 동일한 작업량이 주어지는 경우, 빅코어에서는 작업이 빨리 완료되고 리틀코어에서는 작업이 늦게 완료된다. 이 때, 빅코어가 작업을 완료해도 리틀코어는 여전히 동작 중이기 때문에 화면 단위로 동기화 되어 처리되는 디코딩 작업은 빅코어가 긴 대기시간을 갖게 한다. 반면에 각 코어의 성능비에 맞게 작업을 할당하게 되면 빅코어와 리틀코어는 최대한 동시에 작업을 마칠 수 있으므로 전체적인 작업시간을 줄인다. 본 논문의 연구목표인 이 최적화된 타일 할당은 그림 1과 같다.

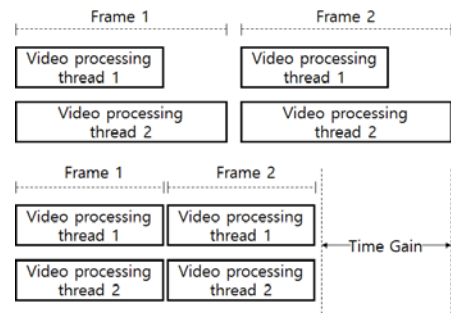


그림 1: 비대칭 코어에 균등하게 타일들을 할당한 경우(위)와 최적화 하여 불균등하게 타일들을 할당한 경우(아래)

비대칭 코어로의 최적화된 타일할당 기법은 각 타일의 복잡도를 예측하는 것이 매우 중요하다. 비대칭 코어의 성능비와 복잡도 정도가 비례하게 할당했다고 하더라도, 그 복잡도 정도가 잘못 예측되었다면 성능개선을 기대하기 힘들기 때문이다. 본 연구에서는 HEVC 예측 유닛인 PU의 분할된 정도(Partitioning)를 이용하는 방법을 함께 제안한다.

2. 관련연구

관련 연구로는 본 연구진의 이전 연구인 비대칭 멀티코어 시스템 상의 HEVC 병렬 디코딩 최적화를 위한 타일의 비대칭 분할 기법[4]이 있다.

비디오 인코딩 단계에서 타일 분할을 수행할 때 비대칭 코어의 성능비와 동일하도록 그림 2와 같이 타일을 비대칭적으로 분할하는 방법이다. 해당 연구는 빅코어 2개와 리틀코어 4개로 구성된 비대칭 멀티코어 CPU의 각 코어마다 1개의 최적화 된 크기의 타일을 할당하여 디코딩을 진행하였고, 약 7%~28%의 디코딩 속도 향상이 있었다. 하지만 이 방식은 (1) 코어의 대칭/비대칭 여부에 따라 다른 비트스트림을 제공해야 하고, (2) 사전에 인코딩 된 영상에는 적용하기 힘든 범용성의 한계가 존재한다.



그림 2: HEVC 공식 테스트 영상(PeopleOnStreet)을 (a)균등한 크기, (b)각 코어의 처리능력을 고려한 타일로 분할한 모습

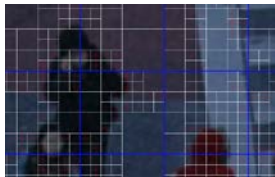


그림 3: PeopleOnStreet 영상의 Coding Unit(CU) 및 PU 분할

3. 비대칭 모바일 코어에 최적화된 HEVC 타일 복잡도 예측 기반 병렬처리

HEVC는 이전 표준화인 H.264/AVC와 유사하게 화면을 여러 개의 블록 단위로 분할하여 영상처리를 한다. 특히, 움직임 예측 (Motion estimation)의 예는 그림 3과 같이 영상이 복잡한 부분은 많이 분할되어 처리되고 그렇지 않은 부분은 덜 분할된다. 가장 큰 단위는 CTU (Coding Tree Unit)이고 각 CTU는 코딩 단위의 CU(Coding Unit)로 분할된 뒤 기본 예측 유닛인 PU까지 다시 한번 분할된다. HEVC의 영상 디코딩 과정에서 ARM 프로세서에서의 각 디코딩 모듈별 처리 시간을 분석 (Profiling)한 논문[5]에 따르면 움직임 예측이 차지하는 정도(Random Access; RA인 경우)는 43%로 상당부분을 차지한다[5]. 그러므로 예측 유닛인 PU가 많을수록 디코딩 시간이 많이 소요되는 복잡한 영상이라고 추정할 수 있다. 따라서, 본 연구는 영상 인코딩 과정에서 별도의 연산 없이 주어지는 PU의 분할된 횟수로 그 영상의 복잡도 (디코딩 시간에 비례) 예측이 가능하다고 판단하고, 먼저 이를 실험한다. PU수와 영상 복잡도의 상관관계를 실제로 측정하기 위해 HEVC공식 Test Sequence 영상 중 하나인 PeopleOnStreet 4K UHD 영상을 RA 코딩 구조의 타일로 분할하여 인코딩한다. 그 후 해당 비트스트림의 각 타일별 PU 분할 정도와 디코딩

시간을 측정하여 회귀분석(Regression)을 통해 상관관계를 분석한다. 그림 4는 각 타일별 PU수와 디코딩 시간을 분산도표로 나타낸 것으로, 실제 PU 분할 정도와 디코딩 시간이 높은 상관관계가 있음을 확인할 수 있다.

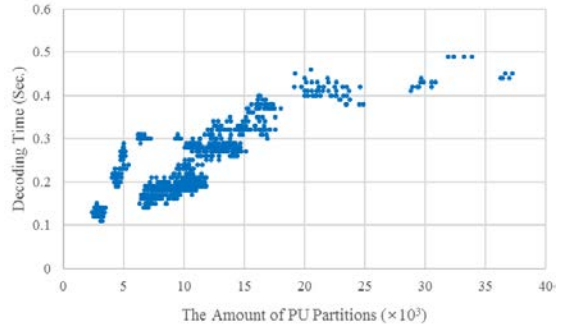


그림 4: PU 분할 정도와 디코딩 시간의 높은 상관관계

본 연구가 제안하는 비대칭 코어에서의 병렬 영상처리 방법은 위의 상관관계 기반의 복잡도 모델을 이용한 2단계를 거친다. 먼저, (1) 각 타일별 PU 분할 정도를 이용하여 각 타일별 복잡도를 높은 정확도로 예측해낸다. 그 후, (2) 예측된 각 타일별 복잡도들을 이용하여 비대칭 코어들의 성능비와 비례하도록 최적화하여 타일들을 코어들에 할당한다. 제안하는 기법은 단일 코어를 사용한 영상처리 뿐만 아니라 비대칭 코어에 순차적으로 타일을 할당하는 현 병렬 영상처리 기술과 비교해도 많은 디코딩 속도 향상의 장점을 가진다.

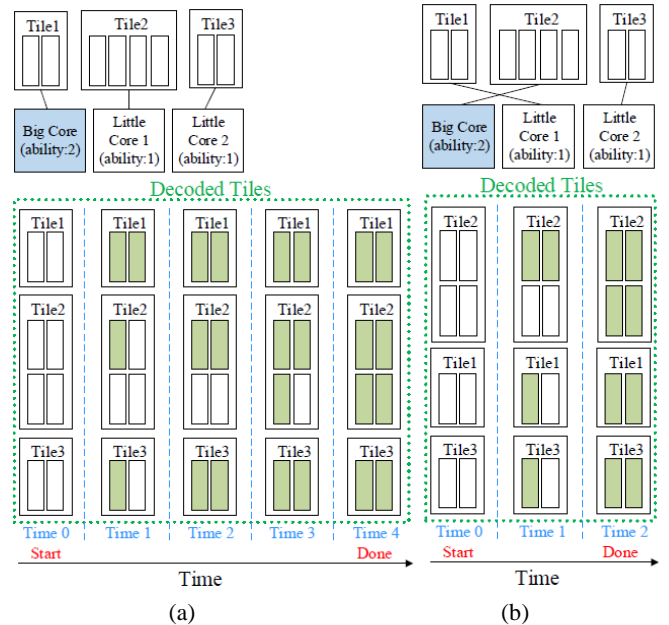


그림 5: (a)현 병렬 영상처리 방법과 (b)제안하는 방법의 타일 할당 방식 및 디코딩에 소요되는 시간의 개념적 설명

그림 5는 위 내용을 도식화한 그림이다. 타일 내부의 네모난 박스는 복잡도 정도를 의미하는데, 빅/리틀 코어의 실제 된 영상 디코딩 속도비가 약 2:1이므로, 빅코어가 한번에 박스 2개를 처리하고 리틀코어는 박스 1개를 처리한다고 본다. 그림 5(a)와 같이 코어에 순차적으로 타일을 할당하면 4개의

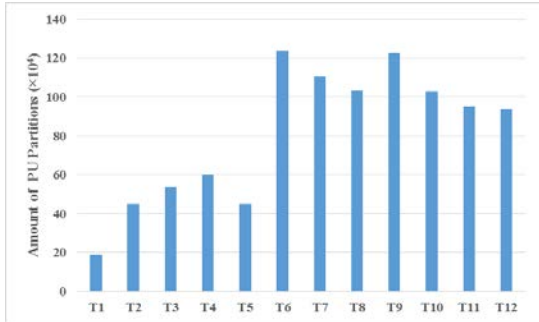
시간 단위(Time 4)가 소모되지만, 제안하는 방식인 그림 5(b)와 같이 비대칭 코어의 성능과 타일의 복잡도를 기반으로 타일을 코어에 할당하면, 시간이 많이 단축되어 (2개의 시간 단위; Time 2) 디코딩 속도 향상을 갖는다.

4. HEVC 타일 기반 병렬 디코더 구현 및 성능 실험

본 실험은 표1의 사양을 갖는 ARM 개발 플랫폼인 JUNO장비를 사용하였으며, HM 15.0 (HEVC reference model)의 인코더 및 디코더를 해당 장비에서 사용하고, HEVC 공식 테스트 영상 중 하나인 *PeopleOnStreet*을 RA 인코딩 옵션과 QP (Quantization Parameter) 22로 인코딩 한다. JUNO장비는 빅코어 2개와 리틀코어 4개로 구성되어 있으므로, 총 코어 개수의 2, 3배수인 12개, 18개의 타일로 영상을 분할하여 실험에 사용한다. 또한, 본 연구는 HEVC 타일 기반의 병렬 디코더를 오픈소스 디코더인 LibDE265 [6]를 기반으로 확장하여 제안하는 기법을 Linux기반의 ARM 플랫폼에 맞춰 구현하였다.

표 1: ARM 개발 플랫폼 JUNO 장비의 사양

big cluster	Cortex-A57 r0p0
LITTLE cluster	Cortex-A53 r0p0
Number of Cores	2 Big cores (1.1GHz) 4 Little cores (850MHz)
Memory	8GB DDR RAM



(a)

B1	B1	B2	B2	L1	L1	L2	L2	L1	L4	L3	L2
L1	L1	L2	L2	L3	B1	B1	B2	L4	B1	B1	B2
L3	L3	L4	L4	B2	L3	L4	L4	B1	B2	B2	L1

(b)

(c)

(d)

그림 6: 각 타일별 PU 분할 정도(a) 및 각 타일별 빅/리틀 코어 할당 그림 (B: 빅코어에 할당, L: 리틀코어에 할당)

그림 6은 12개의 타일로 분할된 경우의 PU 분할 정도(그림 6(a))와 빅/리틀 코어 할당 관계를 나타낸다. 이 때, (1) 실험 대조군인 단순 할당 방식은 그림 6(b)와 같이 타일들을 순차적으로 코어들에 할당한다. 본 연구는 제안하는 기법을 이용하여 (2) 그림 6(c)와 같이 각 코어마다 동일한 수의 타일을 할당해보고, 더 높은 성능 개선을 위해 (3) 그림 6(d)와 같이 각 코어의 성능에 더 적합하도록 할당하는 타일의 수를

최적화 하여 적응적으로 할당한다.

타일 12 및 18개로 나뉜 영상들의 디코딩 속도 향상을 측정한 결과는 표 2와 같다. 제안하는 방식을 사용하여 코어마다 고정된 수의 타일을 할당하는 경우 각각 4.74%와 4.71%의 성능향상이 확인되었다. 또한 제안하는 방식을 사용하여 코어마다 다양한 수의 타일을 적응적으로 할당하는 경우에는 각각 16.84%와 12.93%의 성능향상이 있었다.

표 2: 제안하는 방식으로 얻는 디코딩 속도 성능향상

Tiles	Fixed	Adapted
12	4.74%	16.84%
18	4.71%	12.93%

5. 결 론

본 연구는 비대칭 모바일 코어의 연산 능력을 고려한 새로운 병렬 영상 처리 기법을 제안한다. 제안하는 기법은 HEVC 비디오의 PU 분할 정도를 측정하여 타일 복잡도를 예측하고, 이를 기반으로 각 비대칭 코어들의 성능과 비례하게 최적화하여 타일을 할당한다.

실험을 위해 빅코어 2개와 리틀코어 4개로 구성된 ARM 개발 플랫폼 (JUNO 장비)에서 12개 및 18개의 타일로 분할된 UHD 영상을 사용하였다. 제안하는 기법은 HEVC 오픈소스 디코더인 LibDE265를 확장하여 Linux기반의 ARM 플랫폼에 맞춰 구현되었으며, 전체 디코딩 시간을 측정한 결과, 최대 약 17%의 디코딩 속도 성능향상을 확인하였다.

Acknowledgement

"이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 한국 연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1C1A1A02037743)"

참 고 문 헌

[1] Visual Networking Index Global IP Traffic Forecast, 2015-2020: <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html> (accessed on 21 Oct. 2016)

[2] Misra, K., Segall, A., Horowitz, M., Xu, S., Fuldseth, A., Zhou, M.: An overview of tiles in hevc. *IEEE Journal of Selected Topics in Signal Processing* 7(6), 969-977 (2013)

[3] big.LITTLE Technology, Available online: <https://www.arm.com/products/processors/technologies/biglittleprocessing.php> (accessed on 21 Oct. 2016)

[4] 류영일, 노현준, 류은석 “비대칭 멀티코어 시스템 상의 HEVC 병렬 디코딩 최적화를 위한 타일 분할 기법”, *정보과학회논문지*, Vol. 43, No. 9, pp.1060-1065, Sep. 2016

[5] Bossen, F., Bross, B., Suhring, K., Flynn, D.: Hecv complexity and implementation analysis. *IEEE Transactions on Circuits and Systems for Video Technology* 22(12), 1685-1696 (2012)

[6] LibDE265, Available online: <http://www.libde265.org/> (accessed on 21 Oct. 2016)