

논문 2012-49SD-7-6

멀티코어 시스템에서의 통합된 비디오 디코딩 병렬화

(Integrated Parallelization of Video Decoding on Multi-core Systems)

홍 정 현*, 김 원 진**, 정 기 석***

(Jung-Hyun Hong, Won-Jin Kim, and Ki-Seok Chung)

요 약

고해상도의 동영상 서비스가 보편화 되면서 동영상을 빠르게 처리하기 위한 연구가 활발히 이루어지고 있다. 특히 멀티 코어 시스템 상에서 멀티스레드를 사용한 데이터 레벨 병렬화 방법을 적용하여 비디오 디코더의 성능을 향상시킬 수 있었다. 기존에 제안된 병렬화 방법들을 통해 디코딩 과정의 성능을 향상시킬 수 있었지만, 이 방법들은 엔트로피 디코딩 부분을 제외하거나 엔트로피 디코딩 부분만의 병렬화를 별도로 고려한 부분적인 병렬화 방법이기 때문에 전체 디코딩 과정의 성능 향상에는 부족한 부분이 있다. 따라서 본 논문에서는 기존 병렬화 디코딩 과정뿐만 아니라 엔트로피 병렬화 디코딩 과정까지 함께 고려한 통합적인 비디오 디코딩 병렬화 방법을 제안한다. 우리는 각각의 비디오 디코더 병렬화 방법을 분석하여 최적화 방법을 제시하고 이의 성능평가를 해보았다. 그리고 우리는 비디오 디코딩 과정 내부에 존재하는 코어의 개수에 따른 성능향상의 차이를 고려해 성능을 최적화한 Integrated Parallelization 방법을 제안한다. 우리는 인텔 i7 멀티코어 시스템의 물리적 코어에서 엔트로피 디코딩 부분을 최대로 병렬화 하면서, 내부 자원을 공유하는 하이퍼스레딩 기술을 사용하여 데이터레벨 병렬화 방법에는 물리적 코어 수의 2배까지 스레드를 할당했다. 그리고 디코딩 과정 내부 특성을 고려한 멀티스레드 스케줄링으로 전체 디코딩 과정의 성능을 멀티코어 시스템에 최적화해서 최대 70%까지 성능을 향상시킬 수 있었다.

Abstract

Demand for high resolution video services leads to active studies on high speed video processing. Especially, widespread deployment of multi-core systems accelerates researches on high resolution video processing based on parallelization of multimedia software. Previously proposed parallelization approach could improve the decoding performance. However, some parallelization methods did not consider the entropy decoding and others considered only a partial decoding parallelization. Therefore, we consider parallel entropy decoding integrated with other parallel video decoding process on a multi-core system. We propose a novel parallel decoding method called Integrated Parallelization. We propose a method on how to optimize the parallelization of video decoding when we have a multi-core system with many cores. We parallelized the KTA 2.7 decoder with the proposed technique on an Intel i7 Quad-Core platform with Intel Hyper-Threading technology and multi-threads scheduling. We achieved up to 70% performance improvement using IP method.

Keywords : parallel processing, video decoder

* 정회원, 한양대학교 전자컴퓨터통신공학과
(Department of Electronics and Computer Engineering, Hanyang University)

** 정회원, 삼성전자 DMC 연구소
(Design Solution Lab., DMC R&D Center, SAMSUNG Electronics)

*** 정회원-교신저자, 한양대학교 융합전자공학과
(Department of Electronic Engineering, Hanyang University)

※ 이 논문은 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원 사업의 연구결과로 수행되었음
(NIPA-2012-C1090-1200-0010)

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2010-0024164)과 서울시 산학연 협력사업(JP100031C093211)의 지원을 받아 수행된 것임

접수일자: 2012년1월25일, 수정완료일: 2012년6월21일

I. 서 론

고성능 디지털 장비가 보편화 되고 사용자들은 고해상도의 영상 서비스를 요구하고 있다. 이로 인하여 고해상도 동영상 서비스를 위한 동영상 압축 처리 연구가 활발히 진행되고 있다. 최근 다양한 분야에서 사용되고 있는 H.264/AVC 비디오 부호화 표준은 높은 압축률을 지원하기 위하여 다른 비디오 코덱에 비해 복잡도가 높은 알고리즘을 사용한다. 또한 동영상 압축 국제 표준화 그룹인 MPEG과 VCEG는 기존 압축 기술보다 2배 이상의 압축률을 가진 차세대 동영상 압축 표준인 High Efficiency Video Coding(HEVC)을 연구 중이다.

이처럼 지속적으로 복잡도가 증가하는 비디오 코덱들을 처리하기 위해서는 고성능 프로세서 사용이 필요하다. 기존의 싱글 코어 기반의 프로세서는 클럭 속도를 올려서 성능을 향상시키기 때문에 프로세서의 발열 및 소비전력의 증가로 인하여 성능 향상의 한계에 직면하였다. 이러한 문제를 해결하기 위하여 프로세서 코어 수를 늘리고 병렬 처리를 이용하는 멀티 코어 시스템의 연구가 지속적으로 이루어지고 있다. 그러나 기존의 싱글 코어 프로세서 시스템에서 사용된 프로그램은 멀티 코어 시스템의 성능을 충분히 활용할 수 없다. 따라서 기존의 프로그램을 멀티 코어 시스템에서 효과적으로 실행하기 위하여 병렬프로그램으로 바꾸어야 한다.

멀티코어 시스템에서 영상처리 코딩이 효과적으로 동작할 수 있도록 병렬화하기 위해 다양한 연구들이 진행되고 있다. 여러 가지 병렬화 기법 중 대표적인 데이터 레벨 병렬화 방법은 비디오 데이터를 병렬화가 가능하도록 나누어 여러 스레드에서 동시에 처리하는 방법이다. 데이터 레벨 병렬화 기법은 비디오 데이터의 데이터 의존성을 지키면서 병렬화를 진행해야 한다. 특히 멀티 코어 시스템 상에서 멀티스레드를 사용한 2D-Wave^[10] 방법을 적용하여 엔트로피 디코딩을 제외한 영상처리 디코딩의 성능을 향상 시킬 수 있었다. 엔트로피 부호화 부분은 순차적으로 처리해야 하기 때문에 병렬로 처리하는데 어려움이 있어서 병렬화를 통한 비디오 디코더 성능 향상의 걸림돌이 된다. 특히 고해상도의 영상이 요구되면서 압축률이 높은 Context Adaptive Binary Arithmetic Coding(CABAC) 엔트로피 코딩이 선호되고 있으나, 복잡도가 높아 부호화에서 소요되는 시간이 길다는 단점이 있다. 최근 이러한 문

제를 해결하기 위하여 부호화 성능이 높고 병렬화 성능이 뛰어난 Multi-Threaded Syntax Element Partitioning(MT-SEP) 병렬화 방법^[13]이 제안되었다. 향후 8K-UHD급 고화질 영상처럼 대역폭이 큰 영상들이 상용화 될 예정이기 때문에 고성능 멀티 코어를 사용한 디코딩 병렬화의 성능을 최적화 하는 연구가 지속적으로 요구되고 있다.

기존의 2D-Wave와 MT-SEP 병렬화 방법을 통해 비디오 디코딩 과정의 성능을 향상 시킬 수 있었지만, 기존에 제안된 병렬화 방법들은 엔트로피 디코딩 부분을 제외하거나 엔트로피 디코딩 부분만의 병렬화를 고려한 부분적인 병렬화 방법이기 때문에 전체 디코딩 과정의 성능 향상에는 부족한 부분이 있다. 그래서 우리는 기존의 2D-Wave와 MT-SEP을 분석하고 특성을 파악해 보았다. 그리고 본 논문에서는 엔트로피 디코딩 과정까지 고려한 전체 디코딩 과정의 성능을 최적화하기 위해 Integrated Parallelization (IP) 방법을 제안한다. 우리는 IP 방법을 실제 멀티코어 시스템에서 멀티스레드를 이용하여 구현하여 그 성능을 확인해 보았으며 전체 비디오 디코딩 과정을 효과적으로 병렬화하기 위한 방법을 제시하였다.

본 논문은 다음과 같은 순서로 이루어져 있다. II장에서는 대표적인 비디오 코덱인 H.264/AVC 디코더와 데이터 레벨 병렬화 방법인 2D-Wave 방법 및 최근 발표된 엔트로피 병렬화 알고리즘을 알아본다. III장에서는 본 논문에서 제안하는 IP 병렬화 방법에 대하여 설명한다. 그리고 IV장에서 실험 환경 및 실험 결과에 대해 서술하였으며 논문의 결론과 향후 계획으로 마무리하였다.

II. 관련 연구

1. 영상처리 디코더

H.264 혹은 MPEG-4 AVC는 ISO/IEC와 ITU가 함께 만든 비디오 압축 표준으로 높은 압축률 가지며, 네트워크를 통한 스트리밍에 적합한 특성을 가지고 있어서 다양한 멀티미디어 응용 분야에 많이 쓰이고 있다. H.264/AVC 표준 및 사용되는 알고리즘에 대한 자세한 사항은 ITU의 H.264 표준^[1], ISO MPEG-4/AVC 표준^[2] 문서와 H.264/AVC를 분석한 논문^[3~4]에 잘 기술되어 있다. H.264/AVC 디코더는 Entropy Decoding(ED),

Inverse Transformation(IT), Inverse Quantization(IQ), Intra Prediction(IP), Motion Compensation(MC), Deblocking Filter(DF)로 구성 된다. 간단하게 기능별로 살펴보면 아래와 같다.

가. Entropy Decoding

H.264/AVC에서는 문맥(Context)에 따라 효율적인 엔트로피 코딩이 가능한 형태의 Context-Adaptive Variable Length Coding(CAVLC)과 Context-Adaptive Binary Arithmetic Coding(CABAC) 두 가지의 엔트로피 코딩을 지원한다. 베이스라인 프로파일은 CAVLC, 메인 프로파일과 하이 프로파일은 CABAC를 사용한다.

나. Inverse Quantization/Inverse Transformation

엔트로피 디코딩 작업 후 생성되는 계수(Coefficient) 들은 IQ/IT 과정 후에 residual data로 만들어진다.

다. Intra Prediction/Motion Compensation

IP 과정은 공간적 중복성을 이용하여 화면 내 예측을 하고 MC는 시간적 중복성을 사용하여 화면간의 예측을 수행한다. 그리고 매크로 블록의 타입에 따라서 IP를 적용하거나, MC를 적용한다. IP, MC 후 만들어지는 블록은 IQ/IT 후 에 만들어지는 residual data와 합쳐진다.

라. Deblocking Filter

디코딩 된 영상은 블록간의 경계가 뚜렷이 나타나는 블록킹 현상이 나타난다. 이를 없애기 위해서 블록 경계를 부드럽게 하는 DF를 적용한다. H.264/AVC의 DF는 가중치 값을 제어하면서 적용하는 적응형(Adaptive) 방법을 사용 한다.

2. 영상처리 병렬화 관련 연구

H.264/AVC는 그림 1과 같은 데이터 구조로 이루어져 있다. H.264/AVC의 비디오 시퀀스는 Group Of Pictures(GOP)로 구성된다. 그리고 각 GOP는 여러 개의 픽처(Picture) 또는 프레임(Frame)으로 구성된다. 그리고 하나의 픽처는 하나 또는 여러 개의 슬라이스(Slice)들로 구성된다. 그리고 슬라이스는 여러 개의 매크로 블록(Macroblock)으로 구성되며, 매크로 블록은 휘도 블록(Luma block)과 색차 블록(Chroma block)으

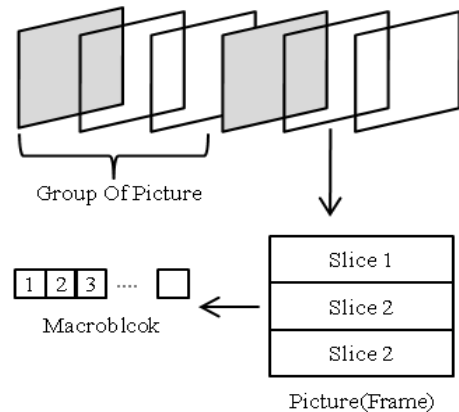


그림 1. H.264/AVC 데이터 구조
Fig. 1. Data Structure of H.264/AVC.

로 이루어진다.

H.264/AVC 디코더를 병렬화 하는데 있어서 앞서 소개된 데이터의 단위에 따라서 다양한 병렬화 방법이 연구 되었다^[5-7]. 데이터 레벨 병렬화 방법은 H.264/AVC 데이터를 병렬화가 가능하게 나누어 처리하는 방법으로 H.264/AVC의 데이터 단위에 따라 프레임, 슬라이스, 매크로 블록 단위 병렬화 방법으로 나누어진다^[8-9].

프레임 단위 병렬화는 참조 프레임에 대한 의존성이 발생하기 때문에 병렬화 성능 향상에 한계가 발생한다. 슬라이스 단위 병렬화의 경우 슬라이스 사이의 데이터 의존성은 없으나 한 프레임을 여러 개의 슬라이스로 나누어 인코딩 된 영상만 슬라이스 단위로 병렬 디코딩이 가능하다. 이러한 이유로 최근 매크로블록 단위의 H.264/AVC 병렬 디코더에 대한 다양한 연구가 이루어지고 있다^[6-11].

매크로 블록 단위 병렬화 방법은 동시에 처리 할 수 있는 매크로 블록을 각 스레드에 할당하여 병렬화 한

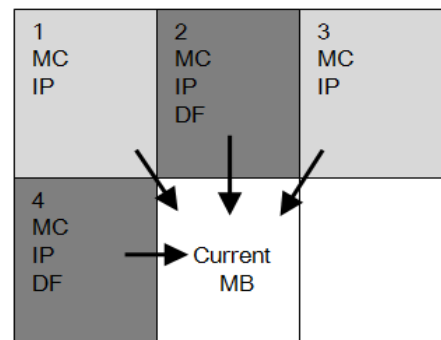


그림 2. 매크로 블록의 데이터 의존성
Fig. 2. Data dependency of Macro block.

다. 하지만 H.264/AVC에서는 그림 2와 같이 매크로 블록 사이의 의존성이 발생 하는데, 예를 들어 그림 2에서 “Current MB”의 IP 처리를 위해서 1, 2, 3, 4 매크로 블록이 먼저 IP 처리 되어야 한다.

대표적인 데이터 병렬화 방법인 2D-Wave 병렬화 방법은 이러한 매크로 블록의 의존성을 고려하여 병렬화 하는 방법이다^[5]. 2D-Wave 병렬화 방법은 그림 3에서 각 화살표에 스투드를 할당하여 화살표를 따라서 매크로 블록을 처리한다. 즉 화살표 방향으로 수평적으로 스투드를 할당하여 병렬화 한다. 예를 들어 그림 3에서 MB(4,0), MB(2,1), MB(0,2)는 데이터 의존성을 지키면서 5번째 시간에서 동시에 처리된다. 이러한 2D-Wave 병렬화 방법을 확장하여 3D-Wave 방법^[10]이 제안 되었다.

그런데 2D-Wave 병렬화 방법에서는 디코딩 단계 전체를 병렬화하지 못한다는 단점이 있다. 데이터 레벨 병렬화를 진행하기 위해서는 ED를 먼저 처리 한다. 그리고 ED를 처리 한 후 MC+IT/IQ IP+IT/IQ를 병렬로 처리한 후 DF를 병렬로 처리한다. ED 과정이 순차 처리되기 때문에 ED 과정을 병렬화 하여 처리 할 수 없다. 그래서 2D-Wave 병렬화 방법은 MC+IT/IQ IP+IT/IQ,DF에서는 효과적인 병렬화 방법이지만 엔트로피 디코더 부분은 추가적인 병렬화 방법이 필요하다.

그리고 영상의 해상도가 올라 갈수록 압축률을 높이기 위해 복잡도가 높아져 엔트로피 디코딩 시간이 길어지는 추세에 있다. 그러므로 엔트로피 디코딩 시간을 줄이기 위하여 병렬화 엔트로피 디코딩의 다양한 연구가 이루어지고 있다.

3. 병렬화 엔트로피 디코딩

H.264/AVC에서는 엔트로피 인코딩 과정에서 CAVLC또는 CABAC 방법을 사용한다. CABAC에서 발생확률은 고정이지 아니라, 지속적으로 업데이트되기 때문에 CAVLC에 비해 더 나은 압축효율을 낼 수 있지만 연산의 복잡도가 훨씬 높아지게 된다. 이러한 특징에 의해 CABAC 연산 과정은 순차적인 구조를 가지게 되어 병렬화를 통한 고속 CABAC 부호화를 수행하는데 어려움이 있다. 이를 극복하기 위해 CABAC를 효과적으로 병렬 처리하기 위한 다양한 연구들이 진행되었다^[12~15]. 하지만 기존 연구들은 전체적인 CABAC 성능 향상에 한계가 있거나 부호화 효율이 떨어지는 단점이 있다.

이러한 CABAC 병렬화에서 발생하는 부호화 효율을 향상 하고 병렬화 성능을 높이기 위하여 Syntax Element Partitioning을 이용한 CABAC 병렬화 방법이 제안 되었다^[14]. Syntax Element Partitioning은 구문요소(Syntax Element)를 그룹으로 나누어 CABAC를 진행하는 방법이다. 구문요소를 MBINFO, PRED, CBP, SIGMAP, COEFF의 그룹으로 나누어 인코딩 한다. 이렇게 인코딩 된 데이터는 Syntax Element 그룹별로 병렬화 디코딩이 가능하다. Syntax Element Partitioning은 기존의 병렬화 방법에 비교하여 코딩 효율을 올릴 수 있다는 장점이 있다.

그런데 CABAC 인코딩 과정에서 SIGMAP 그룹과 COEFF 그룹은 순차적으로 처리되며 그룹데이터의 결속력이 높게 인코딩 된다. 따라서 보다 효율적인 병렬화를 위하여 CABAC를 MBINFO그룹, PRED그룹, CBP그룹 그리고 SIGMAP+COEFF그룹의 4개의 태스크로 나누어 병렬화 하는 Multi-Threaded Syntax Element Partitioning (MT-SEP) 병렬화 방법이 제안되었다^[15]. MT-SEP 방식은 엔트로피 디코딩 과정을 네개의 태스크로 나누어 4단계 파이프라인으로 병렬화를 진행한다. 이때 하나의 Syntax Element를 파이프라인 한 단계에서 처리하는 방식으로 병렬화하면 동기화의

MB(0,0) T1	MB(1,0) T2	MB(2,0) T3	MB(3,0) T4	MB(4,0) T5
MB(0,1) T3	MB(1,1) T4	MB(2,1) T5	MB(3,1) T6	MB(4,1) T7
MB(0,2) T5	MB(1,2) T6	MB(2,2) T7	MB(3,2) T8	MB(4,2) T9
MB(0,3) T7	MB(1,3) T8	MB(2,3) T9	MB(3,3) T10	MB(4,3) T11
MB(0,4) T9	MB(1,4) T10	MB(2,4) T11	MB(3,4) T12	MB(4,4) T13

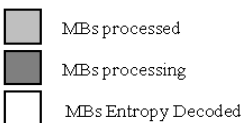


그림 3. 2D-Wave 병렬화 방법
Fig. 3. 2D-Wave parallelization.

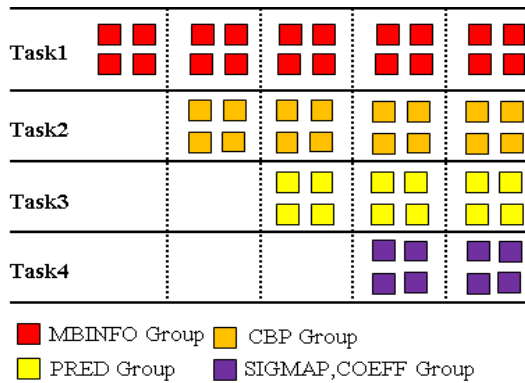


그림 4. MT-SEP 병렬화 방법의 스레드 할당
Fig. 4. Thread allocation for multiple groups of MT-SEP.

표 1. MT-SEP 병렬화 방법의 성능 평가 실험
Table 1. Performance evaluation of MT-SEP.

	before	MT-SEP		MT-SEP		MT-SEP	
	MT-SEP	3 Threads	%	4 Threads	%	5 Threads	%
FHD, 1920X1088	μs	μs	%	μs	%	μs	%
blue_sky	60446	41091	32%	26924	55%	38707	36%
pedestrian area	57079	39180	31%	27768	51%	37608	34%
sunflower	61322	38752	37%	27589	55%	38431	37%
rush_hour	52754	35706	32%	24734	53%	31552	40%

횟수가 많아 전체적인 처리속도가 늦어진다. 이러한 동기화 오버헤드를 줄이기 위하여 그림 4처럼 한 파이프라인 단계에서 Multiple Groups of Syntax Element을 처리하는 방법으로 병렬화가 진행된다.

이처럼 기존에 제안된 2D-Wave와 MT-SEP 병렬화 방법을 통해 비디오 디코딩 과정의 성능을 향상시킬 수 있었다. 하지만 기존에 제안된 병렬화 방법들은 엔트로피 디코딩 부분을 제외하거나 엔트로피 디코딩 부분만의 병렬화를 고려한 부분적인 병렬화 방법이기 때문에 전체 디코딩 과정의 성능 향상에는 부족한 부분이 있다. 따라서 우리는 전체 디코딩 과정의 병렬화를 통합하여 최적화하는 기법을 제시한다.

우리는 인텔 i7 프로세서에서 MT-SEP 엔트로피 디코딩의 병렬화 방법의 성능을 평가했다. 그 결과 표 1에서처럼 MT-SEP 병렬화 방법은 4개의 스레드를 할당 할 경우 성능이 최적화 되는 것을 확인했다. 반면 2D-Wave 병렬화 방법은 그림 5에서와 같이 8개의 논리적 코어를 가지는 인텔 i7 프로세서에서 할당하는 스

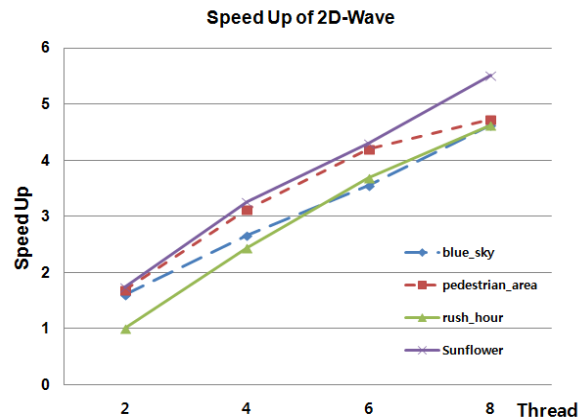


그림 5. 2D-Wave 병렬화 방법의 성능 평가 실험
Fig. 5. Performance evaluation of 2D-Wave.

레드의 수를 2개부터 8개까지 확장할수록 최대 5.5배까지 성능이 좋아지는 결과를 얻을 수 있었다.

이처럼 MT-SEP 병렬화 방법과 2D-Wave 병렬화 방법은 코어의 개수 증가에 따른 성능 향상에서 차이가 존재한다. 그리고 MT-SEP 방법은 병렬화의 단위가 Task 기반으로 이루어지고, 독립된 Syntax Element Partitioning 작업을 스레드에서 분산 실행한다. 2D-Wave 방법은 병렬화의 단위가 Data이기 때문에 순차적인 멀티스레드가 적용된다. 각 병렬화 방법의 특성을 고려하여 성능 평가가 필요하다. 그래서 본 논문에서는 두 방법을 모두 고려한 전체 비디오 디코딩 병렬화 방법의 성능을 평가해 보고 전체 디코딩 관점에서 최적화된 병렬화를 적용한다.

III. Integrated Parallelization

본 논문에서 제안하는 IP 방법의 전체적인 구조는 그림 6과 같다. 첫 번째로 엔트로피 디코딩 부분은 Syntax Element Partitioning 방법을 이용하여 멀티스레드 기반으로 병렬화를 진행한다. 병렬화 엔트로피 디코더를 구현하기 위하여 Syntax Element Partitioning을 파이프라인 방식으로 구현 하였고 OpenMP의 section 기능을 사용해 병렬 구간들을 멀티스레드로 구현 하였다. 두 번째로 MC, IP, IQ/IT, DF 부분은 2D-Wave 병렬화를 적용한다. 2D-Wave 방법은 OpenMP의 parallel for 기능을 이용하여 멀티스레드로 구현 하였다. 이때 MT-SEP와 2D-Wave는 병렬화의

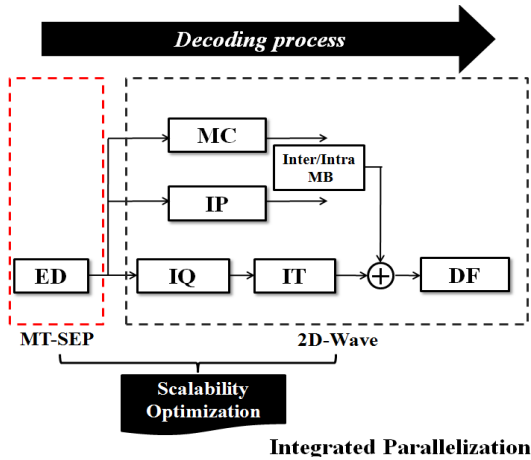


그림 6. Integrated Parallelization의 전체 과정
Fig. 6. Overview of Integrated Parallelization.

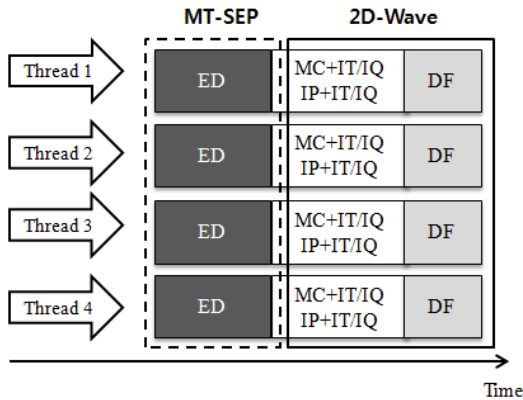


그림 7. Integrated Parallelization의 스레드 할당
Fig. 7. Thread allocation for Integrated Parallelization.

가능성에 차이가 있기 때문에 우리는 두 방법의 코어의 개수에 따른 성능향상도의 균형을 맞추어 전체 디코딩 과정의 성능 향상을 모색 하고자 했다. 마지막으로 병렬화의 단위가 다른 MT-SEP와 2D-Wave방법을 멀티코어 시스템에 최적화하기 위해 OpenMP의 schedule 기능을 이용해 각 디코딩 과정의 특성을 반영한 스레드 스케줄링을 적용했다.

우리는 인텔 하이퍼스레드 기능을 이용하여 디코딩 과정 내부 병렬화 방법들의 가능성의 차이를 고려하여 멀티코어 시스템에 최적화 했다. 먼저 MT-SEP 단계는 4개의 스레드로 병렬화 되었을 때 가장 높은 성능을 내기 때문에 디코딩 과정을 그림 7에서와 같이 4개의 멀티스레드로 병렬화 하여 코어에 할당한다. 이때 2D-Wave 병렬화 기법은 병렬화 가능성이 높기 때문에 4개 이상의 스레드를 코어에 할당해 병렬화 해도 성능

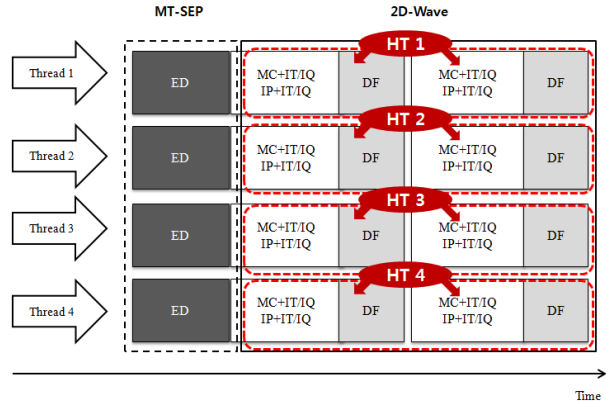


그림 8. Integrated Parallelization의 하이퍼스레딩 할당
Fig. 8. Hyper-thread allocation for Integrated Parallelization.

향상이 가능하다.

따라서 우리는 Intel i7 프로세서에 사용된 동시 멀티스레딩 기법인 하이퍼스레딩 기술을 사용해 2D-Wave 방법에는 물리적 코어의 2배까지 멀티스레드를 할당했다. 우리는 이처럼 병렬화 디코딩 방법들의 병렬화 가능성 차이를 고려하여 최적화하는 방법으로 전체 디코딩 과정의 성능을 향상시켰다.

하이퍼스레딩 기술은 슈퍼스칼라 프로세서에서 동시에 두 스레드로부터 명령어를 가져와 파이프라인에 넣는 기술이다. 하이퍼스레딩 기술을 사용하면 스레드 문맥을 하드웨어 수준에서 복제해서 스레드 수준의 병렬성을 이용할 수 있다. 이때 스레드 문맥 복제를 통해 프로세서의 자원이 공유되기 때문에 추가적인 하드웨어를 구현하지 않고도 병렬화된 어플리케이션들이 프로세서의 내부 자원을 효율적으로 공유할 수 있는 이점이 있다^[16].

따라서 우리는 그림 8에서와 같이 Syntax Element가 MT-SEP병렬화 방법을 통해 4개의 파이프라인 단계로 병렬화 될 동안, 나머지 2D-Wave 방법을 통한 비디오 디코딩 구간은 하이퍼스레딩 기술을 통해 4개의 물리적 코어에 최대 8개까지 멀티스레드를 병렬화 해서 성능을 비교했다.

병렬화의 단위가 다른 MT-SEP방법과 2D-Wave 방법을 전체 디코딩 과정에 적용 후, 성능을 최대화하기 위해서는 각 디코딩 과정의 특성을 고려한 스레드 스케줄링으로 멀티코어 시스템에 최적화 하는 것이 중요하다. 전체 디코딩 과정 중에서 MT-SEP 방법은 병렬화의 단위가 Task 기반으로 이루어지고, 독립된 Syntax

Element Partitioning 작업을 스레드에서 분산 실행한다. 2D-Wave 방법은 병렬화의 단위가 Data이기 때문에 순차적인 멀티스레드가 적용된다.

따라서 우리는 각 디코딩 과정에 적합한 스레드 스케줄링을 실험을 통해 분석했다. 먼저 MT-SEP 방법은 스레드를 Round-Robin 방식으로 정적 할당 하는 것이 최적화된 스케줄링 방식임을 알 수 있었다. 이는 MT-SEP 병렬화 방법은 Syntax element group들이 동기화에 따른 오버헤드를 줄이기 위해 비교적 균등한 크기의 Multiple group들로 묶여서 처리되므로 각 스레드를 균일하게 반복 실행에 적용하는 것이 적합하기 때문이다.

반면 2D-Wave 병렬화 방법은 작업이 먼저 끝나는 스레드에 동적으로 스케줄링을 적용하는 것이 전체적인 성능 향상에 최적화된 스케줄링임을 알 수 있었다. 이는 2D-Wave 병렬화 방법은 처리되는 시간이 균등하지 않은 매크로 블록 단위로 병렬화가 진행되기 때문에, 주변 블록들과의 의존성을 해결한 매크로블록에 먼저

동적으로 스레드를 할당하는 것이 적합하기 때문이다. 따라서 우리는 OpenMP의 schedule 기능을 통해 각 디코딩 과정에 적합한 멀티스레드 스케줄링을 적용하여 전체 디코딩 과정을 최적화 했다.

이처럼 병렬화 비디오 디코더에 최적화를 적용한 IP 방법의 슈도코드는 그림 9와 같다. 먼저 MT-SEP 방법은 멀티스레드를 사용하고 병렬화하고 정적으로 스케줄링 하면서 파이프 라인 방식으로 병렬화가 진행된다. 그리고 2D-Wave 병렬화 방법은 디코딩 과정에 하이퍼스레딩 기술을 적용하여 스레드를 할당하고, 디코딩이 진행되는 반복 횟수를 스레드의 수로 나눈 구간들을 독립적으로 병렬처리 한다. 이때 스레드를 동적으로 스케줄링하여 매크로 블록의 종류에 따라서 디코딩 처리량이 일정하지 않은 2D-Wave 병렬화 방법을 효과적으로 처리 하였다. 자세한 실험 결과는 다음 장에 제시되어 있다.

IV. 실험 환경 및 결과

1. 실험 환경

본 논문에서는 KTA 2.7 디코더를 사용하여 Integrated 병렬화 방법을 적용 하여 실험 하였다. KTA 2.7은 JM 11.0을 기반으로 차세대 동영상에 위한 알고리즘 들이 추가된 소프트웨어이다. 실험을 위해 사용한 영상은 KTA 2.7 에서 제공하는 인코딩 환경 기반으로 인코딩 하였고 quantization parameter(QP)은 26 을 사용하였다. 실험에 사용된 운영체제는 리눅스 Ubuntu 11.04, 커널 버전 2.6.38이다. 실험에 사용된 Intel의 i7-2600 프로세서의 경우 물리적으로는 4개의 코어를 가지고 있지만 Intel의 동시 멀티 스레딩 기술인 하이퍼스레딩을 통해 코어 내부 상태를 복제하여 8개의 논리적 코어로 나타난다^[16].

컴파일러는 cc v4.5.2를 이용하였고 병렬화 방법은 OpenMP^[17]를 사용하였다. 본 논문에서 사용한 OpenMP는 공유메모리 (Shared-memory)구조에서 다중스레드 (multi-thread) 병렬 프로그램을 작성하기 위한 응용 프로그램 인터페이스(API)이다. 사용자는 병렬화가 필요한 부분에 적절한 OpenMP 지시어를 삽입하고 컴파일러는 삽입된 지시어를 참고하여 멀티스레드 코드를 생성한다. 그러므로 OpenMP를 사용하면 쉽게 병렬 프로그래밍을 할 수 있다.

```

01. #pragma omp parallel
02. #pragma omp sections
03. #pragma omp section
04.     FOR each MB in Frame
05.         Do entropy decoding of MBINFO
06.     END FOR
07.
08. #pragma omp section
09.     FOR each MB in Frame
10.         IF not complete MB-1 of MBINFO
11.             Wait
12.         END IF
13.
14.         Do entropy decoding of CBP
15.     END FOR
16.
17. #pragma omp section
18.     FOR each MB in Frame
19.         IF not complete MB-1 of CBP
20.             Wait
21.         END IF
22.
23.         Do entropy decoding of PRED
24.     END FOR
25.
26. #pragma omp section
27.     FOR each MB in Frame
28.         IF not complete MB-1 of CBP
29.             Wait
30.         END IF
31.
32.         Do entropy decoding SIGMAP, COEFF
33.     END FOR
34.
35. #pragma omp parallel for default(shared) schedule(dynamic,1)
36.     FOR each MB in Frame Height
37.         FOR each MB in Frame Width
38.             IF not complete upper and right MB's
39.                 of MC+IQ/IT, IP+IQ/IT and DF
40.                 Wait
41.             END IF
42.             Do MC+IQ/IT, IP+IQ/IT and DF
43.         END FOR
44.     END FOR
    
```

그림 9. Integrated Parallelization의 슈도 코드
Fig. 9. Pseudo code of Integrated Parallelization.

2. 실험 결과

우리는 멀티 코어 시스템에서 엔트로피 디코딩을 제외한 2D-Wave 병렬화 방법, 엔트로피 디코딩 부분만의 MT-SEP 병렬화 방법 그리고 우리가 제안한 IP 방법의 성능을 비교 하였다. 모든 실험결과는 하나의 프

레이에서 하나의 영상을 처리하는 최소시간이다. 표 2는 병렬화 방법 적용 전 프레임 당 디코딩 처리시간이다. 표 3은 엔트로피 디코딩 부분에 MT-SEP 병렬화 방법을 적용해 시간을 측정 한 결과이다. 그리고 표 4는 2D-Wave 병렬화 방법을 적용 후 처리 시간을 측정 한

표 2. 병렬화 이전의 비디오 디코딩 처리 속도
Table 2. Processing speed of video decoding before parallelization.

		ED (μ s)	MC+IT/IQ IP+IT/IQ(μ s)	SUM (μ s)
blue_sky	FHD, 1920X1088	36906	85931	122837
pedestrian_area	FHD, 1920X1088	35722	78088	113810
sunflower	FHD, 1920X1088	38116	97719	135835
rush_hour	FHD, 1920X1088	33691	84374	118065

표 3. MT-SEP 병렬화 방법 적용 후 프레임당 디코딩 처리 속도
Table 3. Processing speed of frame per video decoding after MT-SEP.

ED		Before MT-SEP (μ s)	MT-SEP		SUM	
			(μ s)	(%)	(μ s)	(%)
blue_sky	FHD, 1920X1088	36906	17644	52%	103575	16%
pedestrian_area	FHD, 1920X1088	35722	18045	49%	96133	16%
sunflower	FHD, 1920X1088	38116	17941	53%	115660	15%
rush_hour	FHD, 1920X1088	33691	16128	52%	100502	15%

표 4. 2D-Wave 병렬화 방법 적용 후 프레임당 디코딩 처리 속도
Table 4. Processing speed of frame per video decoding after 2D-Wave.

MC+IT/IQ IP+IT/IQ		Before 2D-Wave (μ s)	2D-Wave (4 Threads)		2D-Wave with Hyper-Threading (8Threads)		SUM	
			(μ s)	(%)	(μ s)	(%)	(μ s)	(%)
blue_sky	FHD, 1920X1088	85931	32950	62%	19795	77%	69856	43%
pedestrian_area	FHD, 1920X1088	78088	30796	61%	18685	76%	66518	42%
sunflower	FHD, 1920X1088	97719	37600	62%	22623	77%	75716	44%
rush_hour	FHD, 1920X1088	84374	31917	62%	19604	77%	65608	44%

표 5. Integrated Parallelization 방법 적용 후 프레임당 디코딩 처리 속도
Table 5. Processing speed of frame per video decoding after Integrated Parallelization.

SUM		Before Parallelization (μ s)	MT-SEP		2D-Wave		Integrated	
			(μ s)	(%)	(μ s)	(%)	(μ s)	(%)
blue_sky	FHD, 1920X1088	122837	103575	16%	69856	43%	40546	67%
pedestrian_area	FHD, 1920X1088	113810	96133	16%	66518	42%	39833	65%
sunflower	FHD, 1920X1088	135835	115660	15%	75716	44%	40564	70%
rush_hour	FHD, 1920X1088	118065	100502	15%	65608	44%	38961	67%

결과이다. 마지막으로 표 5는 IP 방법을 적용 후 기존 병렬화 방법들과의 처리 시간을 비교 한 결과이다.

표 3에서처럼 MT-SEP방법을 적용할 경우 엔트로피 디코딩 부분에서 최대 52%의 성능향상이 있지만 전체 디코딩 관점에서는 최대 16%의 성능향상이 있음을 알 수 있다. 표 4 에서 2D-Wave 병렬화 방법은 엔트로피 디코딩 부분을 제외한 MC+IQ/IT IP+IQ/IT, DF 부분에서는 4개의 물리적 코어에서 최대 66%까지 성능을 향상 할 수 있었다. 또한 하이퍼스레딩 기술을 사용해 최대 8개까지 스레드를 할당할 경우 77%의 성능 향상이 있었다. 하지만 ED 부분이 병렬화 되지 않기 때문에 전체적인 병렬화 성능은 최대 44%까지 향상 되는 것을 알 수 있다. 우리가 제안한 IP 방법은 MT-SEP 병렬화 방법과 2D-Wave 병렬화 방법의 가능성 차이를 최적화 했고 그 결과 표 5와 같이 전체 디코딩 관점에서 최대 70%의 성능 향상이 있었다.

표 5 에서와 같이 전체 디코딩시간을 기준으로 기존에 제안된 MT-SEP 병렬화 방법은 병렬화 되지 않은 비디오 디코딩 방법보다 최대 16% 성능이 향상되었고 2D-Wave 병렬화 방법은 최대 44%의 성능이 향상 되었다. 본 논문에서 제안 하는 IP 방법은 전체 디코딩 관점에서 병렬화가 되기 전보다 최대 70%의 성능 향상이 있음을 확인 할 수 있었다.

IV. 결 론

오늘날 수요가 높아지고 있는 고해상도 영상은 압축 기술의 복잡도와 데이터의 양이 계속해서 증가하고 있는 추세이기 때문에 비디오 디코더의 병렬화를 통한 영상압축 기술의 성능 향상은 지속적으로 연구되어야 한다.

기존의 비디오 디코더 병렬화 연구들은 효과적으로 CABAC를 병렬화하기 위하여 태스크 단위의 MT-SEP 병렬화 방법을 사용하였고, 이 과정에서 병렬화 시 발생하는 동기화 오버헤드를 줄이는 방법을 제시했다. 나머지 영상처리 부분은 대표적인 데이터 단위 병렬화 알고리즘인 2D-Wave를 적용하여 병렬화를 진행하였다. 기존의 병렬화 연구들은 엔트로피 디코딩과 그 외의 디코딩 과정의 성능 향상 목표로 했기 때문에 전체적인 디코딩 과정의 성능 향상에는 부족한 부분이 있었다.

따라서 본 논문에서 제안한 Integrated Parallelization

(IP) 방법은 비디오 디코딩 과정 전체를 병렬화 하는 것을 목표로 했다. 특히 MT-SEP 병렬화 방법은 4개의 코어를 사용한 병렬화에 최적화 되어 있지만, 2D-Wave 병렬화 방법은 4개 이상의 코어를 사용해도 병렬화 성능이 증가하는 코어의 개수에 따른 성능향상의 차이가 있었다. 그러므로 우리는 각 디코딩 과정들의 코어의 개수에 따른 성능향상의 차이를 고려해 전체 디코딩 과정의 성능을 최적화 했다.

우리는 인텔 i7 멀티코어 시스템의 4개의 물리적 코어에서 엔트로피 디코딩 부분을 최대로 병렬화 하면서, 내부 자원을 공유하는 하이퍼스레딩 기술을 통해 2D-Wave 방법에 8개까지 스레드를 할당했다. 그리고 각 디코딩 과정의 특성을 반영한 스레드 스케줄링을 적용하는 확장된 병렬화 방법으로 전체 디코딩 과정의 성능을 최적화 했다. 이를 기반으로 멀티 코어 상에서 멀티스레드를 사용하여 실험한 결과 병렬화 되지 않았을 때 보다 MT-SEP 병렬화 방법은 16% 성능이 향상되었고 2D-Wave 병렬화 방법은 최대 44%의 성능이 향상 되었다. 본 논문에서 제안 하는 IP 방법은 최대 70%의 성능 향상이 있음을 확인 할 수 있었다.

향 후 연구로 본 논문에서 실험에 사용된 시스템뿐만 아니라 다양한 멀티 코어 시스템에서 IP 방법을 적용하여 각 시스템에서의 성능 차이를 비교 분석하고, 해당 시스템에서 최적의 결과를 내는 병렬화 방법을 모색할 계획이다.

참 고 문 헌

- [1] ITU-T Recommendation H.264, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services- Coding of moving video, May 2003.
- [2] ISO, Information Technology-Coding of Audio-Visual Objects, Part10-Advanced Video Coding, ISO/IEC 14496-10.
- [3] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, Senior Member, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003
- [4] Michael Horowitz, Anthony Joch, Faouzi Kossentini, and Antti Hallapuro, "H.264/AVC Baseline Profile Decoder Complexity Analysis,"

- IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 704-716 July 2003.
- [5] E. van der Tol, E. Jaspers, and R. Gelderblom, "Mapping of H.264 decoding on a multiprocessor architecture," *Image and Video Communications and Processing*, pp.707-718, May, 2003.
- [6] J. Chong, N. R. Satish, B. Catanzaro, K. Ravindran, and K. Keutzer, "Efficient parallelization of h.264 decoding with macro block level scheduling," in 2007 IEEE International Conference on Multimedia and Expo, July 2007.
- [7] Kosuke Nishihara, Atsushi Hatabu, Tatsuji Moriyoshi, "Parallelization of H.264 video decoder for embedded multicore processor," In Proceedings of ICME'2008. pp.329~332
- [8] 심동규, 남정학, "고속 비디오 처리를 위한 병렬화 기술," 대한전자공학회, 전자공학회논문지, 제36권 제4호 (통권 제299호), 83-90쪽, 2009년 4월
- [9] 조한욱, 조송현, 송용호, "멀티코어 프로세서에서의 H.264/AVC 디코더를 위한 데이터 레벨 병렬화 성능 예측 및 분석," 전자공학회논문지, 제46권, 제8호 102-116쪽, 2009년 8월
- [10] A. Azevedo, C. Meenderinck, B. Juurlink, A. Terechko, J. Hoogerbrugge, M. Alvarez, and A. Rammirez, "Parallel H.264 Decoding on an Embedded Multicore Processor," in Proceedings of the 4th International Conference on High Performance and Embedded Architectures and Compilers -HIPEAC, Jan 2009.
- [11] Won-Jin Kim, Keol Cho, Ki-Seok Chung, "Stage-based frame-partitioned parallelization of H.264/AVC decoding," *IEEE Trans. on Consumer Electronics*, vol. 56, pp. 1088 - 1096, May. 2010.
- [12] Chung-Hyo Kim, In-Cheol Park, "Parallel Decoding of Context-Based Adaptive Binary Arithmetic Codes Based on Most Probably Symbol Prediction," *IEICE Trans. on Information and Systems*, vol. E90-D, no. 2, pp. 609 - 612, February 2007.
- [13] Andrew Segall, Jie Zhao, "Entropy slices for parallel entropy decoding," ITU-T SGI 6/Q.6 Doc. COM16-C405. Geneva, Switzerland, April 2008.
- [14] Vivienne Sze, Anantha P. Chandrakasan, "A High Throughput CABAC Algorithm Using Syntax Element Partitioning," *IEEE International Conference on Image Processing (ICIP)*, pp. 773-776, November 2009.
- [15] Won-Jin Kim, Keol Cho, Ki-Seok Chung, "Multi-Threaded Syntax Element Partitioning for Parallel Entropy Decoding," *IEEE Trans. on Consumer Electronics*, vol. 57, pp.897 - 905, May. 2011.
- [16] Deborah T. Marr et al. "Hyper-Threading Technology Architecture and Microarchitecture", *Intel Technology Journal Q1*, 2002.
- [17] Chunhua Liao, Zhenying Liu, Lei Huang, and Barbara Chapman. "Evaluating OpenMP on Chip MultiThreading Platforms," In First international workshop on OpenMP, Eugene, Oregon USA, June 2005. April 2004.

저 자 소 개



홍 정 현(정회원)
2011년 한양대학교 미디어통신
공학과 학사 졸업.
2012년 현재 한양대학교
전자컴퓨터공학과
석박사 통합과정.

<주관심분야: 영상처리, 임베디드 시스템, 멀티코어 시스템>



정 기 석(정회원)-교신저자
1989년 서울대학교 컴퓨터공학과
학사 졸업.
1998년 Univ. of Illinois at
Urbana-Champaign,
Computer Science
박사 졸업.

1998년 Univ. of Illinois at Urbana-Champaign,
강의 전담 교수.

2000년 Synopsys, Inc. Sr. R&D Engineer.

2001년 Intel Corp. Staff Engineer.

2001년 홍익대학교 컴퓨터 공학과 조교수.

2004년 한양대학교 정보통신대학 조교수.

현재 한양대학교 융합전자공학부 부교수.

<주관심분야: 임베디드 시스템, VLSI 및 SoC 설계>



김 원 진(정회원)
2002년 한양대학교 주전공 기계공
학과, 부전공 전기전자컴
퓨터공학과.

2004년 세나소프트 기술연구소
연구원.

2006년 (주)엠게임 개발팀 연구원.

2008년 한양대학교 전자컴퓨터통신공학과 석사.

2012년 한양대학교 전자컴퓨터통신공학과 박사.

현재 삼성전자 DMC 연구소 Design Solution
Lab.

<주관심분야 : 멀티코어 시스템, 영상처리, 임베
디드 시스템, SoC 설계>