

# HEVC 화면 크기별 비디오 디코딩 복잡도에 관한 분석

노현준, 류영일, 류은석  
가천대학교 컴퓨터공학과

e-mail : ggyo@gc.gachon.ac.kr  
wlrmlrm99@gc.gachon.ac.kr  
esryu@gachon.ac.kr

## Video Decoding Complexity Analysis Based on HEVC Resolution

Hyun-Joon Roh, Yeongil Ryu, Eun-Seok Ryu  
Department of Computer Engineering, Gachon University

### 요 약

본 논문은 JCT-VC 의 표준 레퍼런스 소프트웨어인 HM15.0 을 통해서 HEVC 비디오를 다양한 크기와 모드로 인코딩한 뒤 동일한 워크스테이션에서 디코딩 작업을 수행한다. 이 결과로 나오는 디코딩 시간을 토대로 비디오 크기별 디코딩 복잡도에 관한 다중회귀분석을 진행한다.

### 1. 서론

최근 UHD(Ultra High Definition) (초고해상도 4K) 이상의 영상을 지원하는 시스템이 활발하게 보급되고 있다. 이에 따라 2013 년 초 국제 비디오 코딩 표준화 단체인 ISO/IEC MPEG 과 ITUT VCEG 가 공동으로 조직한 JCT-VC(Joint Collaborative Team for Video Coding)에서 새로운 HEVC(High Efficiency Video Coding) 비디오 압축 기술을 표준화하고 이에 대한 연구를 진행한다. 모바일 디바이스 역시 이런 변화에 적응하여 UHD 영상을 재생 가능하도록 개발된다.

최근 출시되는 모바일 디바이스들은 시장의 요구에 따라 화면이 커지는 데에 반해 두께는 더 얇게 출시된다. 이에 따라 배터리를 많이 소모하게 되고, 배터리 용량은 줄어들게 된다. 이 때문에 배터리의 높은 성능이 요구되는데 배터리 성능 발전의 속도는 리튬이온전지 개발 이후에 더디다. 따라서 모바일 디바이스에서는 다른 기기들보다 배터리를 얼마나 효율적으로 소모하는가 신경을 써주어야 한다.

이러한 효율적인 배터리 소모를 위해서는 여러 가지 방법이 있는데 그것들 중에 하나가 CPU 자원을 수행하는 작업에 적합하게 할당하는 것이다. 디코딩 역시 CPU 를 활용하는 작업이기 때문에 자원 할당에 신경 써야 한다. 하지만 현재 사용하고 있는 병렬 디코딩 방식은 화면의 크기나 복잡도에 상관없이 자원을 할당한다.

따라서 본 연구는 모바일 디바이스에서 병렬 디코딩을 할 때 기존의 방식과는 다른 적응적 비디오 디코딩 및 병렬 처리를 제안한다. 또한 이를 적용하기 위해 JCT-VC 에서 HEVC 비디오 압축 기술을 연구하

기 쉽도록 제공하는 HM (HEVC reference Model) 소프트웨어를 이용하여 비디오의 크기 등으로 디코딩 복잡도가 어떻게 나타나는지 알아보고 이를 통한 다중 회귀분석을 진행한다.

### 2. 관련연구

#### 2.1. JCT\_VC HEVC 비디오 표준

2010년 1월 MPEG 과 VCEG 은 공동으로 새로운 동영상 압축 표준 기술을 개발하기로 하였다. 이에 따라 만들어진 협력 팀이 JCT-VC 이고 공동으로 최종 버전의 Joint CfP(Call for Proposal)를 공지하였다. 2010년 4월 독일의 Dresden 에서 첫 JCT-VC 미팅이 열렸는데 여기서 Joint CfP 의 응답으로 총 27 개의 기술이 제안되었다. 이러한 기술들 중 우수한 기술들을 사용한 초기 테스트 모델 작업이 진행되었고 그 결과로 선정된 초기 모델을 바탕으로 2015년 10월까지 총 22번의 JCT-VC 미팅을 통해 차세대 비디오 코덱 기술의 개발 및 표준 제정을 진행하였다[1].

#### 2.2. 영상 프레임 디코딩 복잡도 예측 기술

디코딩은 일정한 시간마다 화면에 출력해야 하므로 실시간성을 갖는다. 특히 중요한 것은 절대적 데드라인 시간이 아닌 상대적인 시간이다. 이전의 방식은 영상입력시점과 출력시점이 고정되어 있다고 가정하거나 디스플레이 버퍼를 제대로 활용하지 못했다는 것에 주목한 한 연구에서는 새로운 DVFS (Dynamic Voltage and Frequency) 알고리즘을 제시한다. 일단 영상 시퀀스가 시간에 따라 변화하는 것 때문에 전체 통계 데이터를 적용해서 예측하는 방법은 어렵다는 것을

인지하고, 예측하는 방법으로 슬라이딩 윈도우를 사용한다. 이는 시간적으로 가까운 화면의 내용을 예측에 사용 가능하게 해준다. 또한 프레임 타입마다 디코딩 복잡도 상관관계가 다르다는 것을 인지하고 프레임들에 각 타입마다 다른 예측수치를 적용한다. 따라서 모든 프레임은 각 타입으로 구분하고, 구분한 각 타입별 선형모델로 프레임의 계산량을 예측한다. 이런 작업들을 통해 디코딩을 위한 자원을 분배하여 전력의 소모를 줄이는 것이다[3][4].

하지만 이 연구는 연산단위를 한 프레임으로 한정하여 예측하였고, 영상 해상도 및 사용 비디오 코덱도 오늘날의 기술과는 차이가 있으므로 본 연구에 직접 적용은 무리가 있다.

### 3. 비디오 해상도에 따른 디코더 복잡도 분석

앞서 언급한대로 모바일 디바이스는 CPU의 자원 할당 최적화를 통해 배터리의 소모를 줄여야 하는 제한적인 상황 속에 놓여있다. 디코딩 최적화도 이러한 자원 할당 최적화 작업 중 하나이고 이는 곧 배터리 소모감소로 이어진다. 본 연구는 디코딩 최적화를 위한 방안으로 적응적 비디오 디코딩 및 병렬 처리를 제안한다.

이를 위해서는 어떠한 규격으로 나누어야 가장 효율적인지 알아야 한다. 따라서 각 옵션의 다양한 규격의 영상을 디코딩하여 규격마다의 자원 소모 정도를 알아내고 이를 수치화한다. 이러한 실험을 통해 얻어낸 효율적인 규격에 대한 데이터를 토대로 차후에 디코더의 디코딩 방식을 개선한다. 구체적인 방안은 타일(Tile) 또는 슬라이스 단위로 화면을 분할하여 병렬 디코딩하는 방식이다.

#### 3.1. 분석실험

실험환경은 HEVC 비디오 압축 기술을 활용하기 위해 앞서 언급한 HM의 버전 중 하나의 HM15.0을 이용하였다. 사용한 영상의 경우 HEVC 표준화를 위한 테스트 영상들 중 하나인 PeopleOnStreet (3840×2160, 30 frame/sec)(그림 1)을 활용한다. 디코딩의 경우 보유한 워크스테이션(Intel Zeon CPU core 12개, hyper-thread 24개, RAM 48GB)을 통해 진행한다.

본 연구에서는 동일한 구동환경에서 동일한 디코더로 서로 다른 옵션으로 인코딩된 비트스트림을 디코딩하며, 이때 측정된 디코딩 시간을 토대로 해당 영상의 복잡도를 판단한다[5].

표 1은 연구를 진행하면서 인코딩시 사용한 옵션 값들을 나타낸 표이다. 이렇게 3가지 옵션들을 변경해가며 인코딩하여 얻은 84가지 비트스트림들을 동등한 조건에서 디코딩한다. 디코더의 경우에는 인코더와 달리 별도의 옵션을 추가하지 않고 디코딩을 실시한다. 최종적으로 이 결과값을 토대로 분석을 실시한다.



(그림 1) UHD 실험영상 (PeopleOnStreet 3840×2160)

<표 1> 인코딩시 사용한 옵션 값 목록

Sequence		PeopleOnStreet	
Coding Status	Random Access (RA)		384×216
	All Intra (AI)		640×360
	Low-Delay P (LDP)		896×504
QP	22	Resolution	1280×720
	27		1920×1080
	32		2560×1440
	37		3840×2160

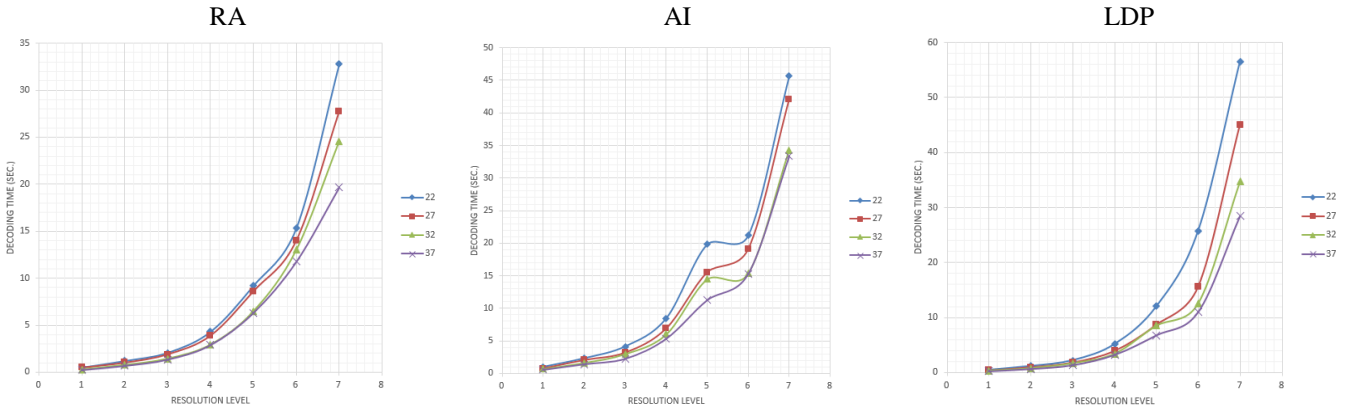
#### 3.2. 실험결과

실험결과는 표 2와 같이 보여진다. 그림 2는 표 2를 각 모드별로 보기 쉽게 나타낸 그래프이다.

총 7단계의 해상도로 나누어지는데, 각 단계의 넓이를 구하여 각 단계별로 증가하는 넓이 비를 구해보면 가장 낮은 해상도부터 각각 약 2.78, 1.96, 2.04, 2.25, 1.77, 2.25와 같이 대략 2배 정도의 증가 정도를 보인다. 따라서 디코더의 복잡도 역시 각 단계마다 2배 정도씩 증가하지 않을까 예상할 수 있다. 그런데 그림 2를 통해 증가되는 모습을 각 모드별로 살펴보면 모든 모드들의 전체적인 모습이 정비례가 아닌 제곱비례의 형태를 나타낸다. 이는 디코더의 복잡도가 단순히 해상도의 넓이 비와 정비례하는 것이 아니라 각 규격마다 복잡도 증가 정도가 다르다는 것을 보여주고, 이에 따라 자원의 할당에 조절이 필요하다.

또한 같은 해상도인 경우에도 모드가 다르면 디코딩 시간이 다르게 나타나므로 해상도뿐만 아니라 인코딩 시에 사용했던 모드 역시 디코더의 복잡도에 영향을 준다는 것을 보여준다. 따라서 디코더는 인코더의 인코딩 모드에 따라 다른 자원 할당 방법을 도입해야 한다.

특히 해상도 7단계(3840x2160)의 경우 복잡도의 가장 큰 증가 폭을 보여주며 이는 UHD 이상의 영상의 경우에는 더욱 효율적인 디코딩 방식이 연구되고 도입되어야 한다는 것을 보여준다.



(그림 2) 해상도별 디코딩 시간 측정 그래프 (RA: Random Access, AI: All Intra, LDP: Low-delay P)

CPU의 발전에 따라 최근의 모바일 디바이스들은 보통 둘 이상의 코어(Core)를 가진다. 이를 이용해서 병렬 디코딩을 실시하면 더욱 빠른 디코딩을 실시할 수 있는데 이는 쉬는 코어를 제거함으로써 빠른 디코딩 완료로 이어지고 전력소모 감소로 이어진다. 이때 모든 코어들이 디코딩 작업하기 시작해서 동시에 종료되면 쉬는 코어가 한 순간도 없게 되므로 가장 이상적인 상황이다. 따라서 영상을 분할하여 코어에 분배해줄 때 각 코어의 성능의 비와 적합하게 영상을 분배해줌으로써 종료시간을 최대한 일치시킨다. 이때 앞서 조사했던 인코딩 모드별, 그리고 해상도별 디코더 작업 복잡도 수치가 활용된다. 이러한 작업을 수행할 때 매 프레임마다 분할해주어야 하므로 타일이나 슬라이스 단위로 분할해준다.

<표 2> 해상도별 디코딩 시간 측정

Coding structure	Resolution	Decoding time / QP			
		22	27	32	37
RA	384x216	0.493	0.482	0.303	0.231
	640x360	1.191	1.014	0.773	0.678
	896x504	2.067	1.894	1.445	1.344
	1280x720	4.299	3.887	2.938	2.887
	1920x1080	9.203	8.609	6.488	6.314
	2560x1440	15.321	14.01	13.026	11.762
	3840x2160	32.795	27.76	24.586	19.667
AI	384x216	0.998	0.764	0.724	0.55
	640x360	2.31	2.039	1.661	1.414
	896x504	4.132	3.182	3.011	2.252
	1280x720	8.448	6.907	6.024	5.338
	1920x1080	19.809	15.518	14.459	11.27
	2560x1440	21.169	19.094	15.366	15.303
	3840x2160	45.677	42.12	34.304	33.384
LDP	384x216	0.456	0.379	0.319	0.237
	640x360	1.153	0.974	0.756	0.656
	896x504	2.2	1.823	1.658	1.347
	1280x720	5.138	3.945	3.384	3.219
	1920x1080	12.015	8.694	8.497	6.774
	2560x1440	25.787	15.615	12.542	10.998
	3840x2160	56.566	45.037	34.726	28.423

<표 3> 다중회귀분석결과

Coding structure	다중회귀분석 식	결정계수
RA(Random Access)	$y=3.358 \times 0.967^{x^2} \times 1.037^{x^1}$	0.719
AI(All Intra)	$y=6.250 \times 0.969^{x^2} \times 1.033^{x^1}$	0.700
LDP(Low-delay P)	$Y=4.072 \times 0.961^{x^2} \times 1.041^{x^1}$	0.751

#### 4. 결론

본 연구는 비디오 해상도와 비디오 디코딩 복잡도 사이의 상관관계를 실험을 통해 분석하였다.

본 논문은 JCT-VC의 표준 테스트 조건(CTC: Common Test Condition)에 의한 실험을 통해 비디오 해상도 다중회귀분석결과 표 3과 같이 표현 가능함을 밝혔다.

이는 향후적으로 비디오 해상도에 따른 병렬 디코딩에 응용될 수 있다. 특히 모바일 디바이스와 같이 한정된 자원을 효율적으로 분배해야만 하는 곳에서는 매우 의미있는 기술이며, 현재 비대칭 코어(Asymmetric Cores)를 이용한 고화질 비디오 병렬처리에 관하여 연구 중이다.

#### Acknowledgement

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

#### 참고문헌

- [1] 심동규, 조현호, "고효율 영상 부호화 기술 HEVC 표준 기술의 이해", 홍릉과학출판사, Jan. 2014.
- [2] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol.22, no.12, pp.1649,1668, Dec. 2012.
- [3] 안희준, 정승호, "영상 프레임 디코딩 복잡도 예측을 통한 DVFS 전력감소 방식," 한국통신학회논문지 '13-01, vol.38B, no.01, pp.46-53, Jan. 2013
- [4] A.C. Bavier, A. B. Montz, Larry and L.Peterson, "Predicting MPEG execution times," SIGMETRICS Perform. Eval. Rev. 26, 1, 131-140, Jun. 1998.

- [5] Bossen, F.; Bross, B.; Suhring, K.; Flynn, D., "HEVC Complexity and Implementation Analysis," IEEE Transactions on Circuits and Systems for Video Technology (CSVT), vol.22, no.12, pp.1685-1696, Dec. 2012.