

# HEVC 엔트로피 부호화 기술 동향

유은경·심동규 (광운대학교)

## I. 서론

현대 사회는 높은 기술력과 첨단 인프라 구축으로 인해 멀티미디어 서비스가 대중화 되면서 멀티미디어는 많은 사람들이 정보를 얻고 즐거움을 느끼는 수단으로 자리 잡았다. 이러한 멀티미디어 서비스를 보다 높은 질로 서비스 하고, 언제 어디서나 서비스가 가능하도록 하기 위해 많은 제품들이 개발되고 있다. 멀티미디어 서비스를 제공하는 모바일 기기, 디스플레이 장치들은 소형화 및 대형화 추세에 따라 각 기기에 알맞은 높은 해상도를 지원하거나 낮은 복잡도를 갖는 압축 기술의 필요성이 증대되고 있다.

이러한 사회적 흐름에 따라 비디오 압축 표준화 단체인 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)에서는 현재 비디오 압축 표준에서 최신 압축 기술인 H,264/AVC 보다 두 배의 압축 효율 또는 같은 압축 효율에서의 낮은 복잡도를 목표로 공동 표준화 단체인 JCT-VC(Joint Collaborative Team on Video Coding)를 구성하여 2010년 4월 독일 드레스덴 회의를 시작으로 차세대 비디오 코딩 표준인 HEVC(High Efficiency Video Coding)의 표준화를 진행하고 있다. HEVC 표준화 과정을 간단히 살펴보면 다음과 같다. 2010년 4월 독일 드레스덴에서 열린 JCT-VC 표준화 회의에서는 각 기관의 비디오 코덱을 비교 및 평가를 통하여 상위 7개를 결정하고 이를 재구성하여 TMuC(Test Model under Consideration)를 발표했다. 이 후, 2010년 10월 중국 광저우 회의에서는 TMuC에 구현된 기술 중 각 툴(tool) 별로 압축 효율 대비 낮은 복잡도를 가지는 기술을 채택하여 HM(HEVC Test Model)이라는 레퍼런스 소프트웨어를 공표하였다. 이 후 진행된 표준화 회의인 2011년 1월 한국 대구 회의와 같은 해 3월 스위스 제네바 회의에서는

표준으로 결정된 기술 중 논의할 내용을 CE(Core Experiment)로 결정하고 다음 표준화 때 CE 별로 제안된 기술을 비교 평가하여 표준으로 채택하고 있다.

본 기사에서는 HEVC의 다양한 기술들 중 엔트로피 부호화 방법에 대해서 소개한다. 현재 HEVC에서 엔트로피 코딩으로 채택된 기술은 인코딩 모드에 따라 두 가지로 분류하는데, 저복잡도 모드에서 사용되는 CAVLC와 고효율 모드에서 사용되는 CABAC이다. 두 엔트로피 코딩 방법은 H,264/AVC에서 사용한 엔트로피 코딩 기술과 유사한 형태지만 HEVC의 CAVLC의 경우 기존 H,264/AVC의 CAVLC와 구문(syntax) 종류 및 테이블 업데이트 방법이 다르며 CABAC의 경우 기본 구조는 같지만 문맥 모델 방법 등에 있어 현재 HEVC 코딩 구조에 적합한 형태로 조정되었다.

본 기사의 구성은 다음과 같다. 2장에서는 기존 동영상 부호화에 사용된 엔트로피 코딩 방법에 대해 간단히 알아본다. 3장에서는 HEVC 엔트로피 부호화에 대하여 살펴본다. 마지막 4장에서는 향후 표준화 방향을 전망하며 결론을 맺는다.

## II. 엔트로피 부호화 기술

본 장에서는 종래의 엔트로피 부호화 기술에 대해서 소개한다. 엔트로피 부호화 기술에 대해 알아보기 전에 엔트로피의 정의에 대해 먼저 살펴본다. 엔트로피는 열역학 제 2법칙에서 어원을 두고 있으며, 엔트로피의 개념은 이후 여러 분야에서 다양한 의미로 사용되고 있다. 그 중 정보이론 분야에서는 어떤 정보를 표현하는데 필요로 하는 최소 정보량을 의미한다. 정보이론에서의 엔트로피에 대해 좀 더 자세히 설명하면, 보내고자 하는 정보를 표현하기 위한 기본 단위인 심볼

들에 대해 각각 발생확률을 알고있다고 가정한다. 각각의 발생확률을 이용하여 정보량 측정함수로 최소의 정보량을 구할 수 있다. 모든 심볼에 대해 정보량을 구하고 이를 발생확률에 따라 곱하여 평균을 구하면, 그 확률 변수에 대한 평균 정보량을 얻을 수 있다. 이때 얻어진 평균 정보량을 엔트로피라 하며 실제 확률에 의해 얻어진 값이므로 최소 평균 정보량이 된다.

엔트로피 코딩 기술은 무손실 압축 기법으로 부호화 할 심볼에 대해서 동일한 길이의 비트를 할당하지 않고 발생확률에 따라 가변길이의 비트를 할당하는 것이다. 이러한 엔트로피 코딩 기법은 심볼의 발생확률에 기초하여 비디오 압축률을 높이기 위하여 사용되는데, 방법은 다음과 같다. 비디오 압축 후 출력된 구문을 기본 단위로 하여 구문에서 출력될 수 있는 모든 값을 심볼이라 할 때, 모든 심볼에 대해서 심볼의 발생확률이 높은 순부터 차례대로 나열한다. 비디오 코딩의 경우 발생확률이 어느 정도 예측 가능하기 때문에 발생확률이 높은 심볼인 경우 짧은 비트를 할당하고 발생확률이 낮은 심볼인 경우 긴 비트를 할당하여 출력으로 보내야 하는 평균 정보량을 줄인다.

엔트로피 코딩 기법은 크게 두 가지로 분류한다. 가변길이 코딩(VLC: Variable Length Coding)과 산술 코딩(AC: Arithmetic Coding)이다. H.264/AVC 엔트로피 코딩 이전의 기술은 주로 간단한 가변길이 부호화 방법을 사용하였다. 하지만 산술 코딩의 높은 압축효율 사용을 위해 H.264/AVC는 가변길이 코딩 기반의 적응적 가변길이 코딩(CAVLC: Context-based Adaptive Variable Length Coding)과 산술코딩 기반의 문맥 기반 적응적 산술코딩(CABAC: Context-based Adaptive Variable Length Coding)을 채택하였다. H.264/AVC의 두 엔트로피 코딩 기법의 공통점은 심볼의 발생확률을 좀 더 정확하게 측정하기 위해 주변 문맥 정보를 이용하고(context-based) 적응적인(adaptive) 확률 예측을 수행한다는 것이다. 이는 엔트로피 코딩의 기본단위인 심볼들이 영상의 특성에 따라 발생확률이 변화하므로, 각 심볼에 대한 발생확률을 정확하게 얻기 위함이다. 여기서 심볼의 발생확률을 정확하게 예측할 경우, 최소의 평균 정보량인 엔트로피에 근접하게 된다.

지금까지, 엔트로피의 정의와 엔트로피 코딩이 동영상에 쓰이는 기본적인 방법에 대하여 알아보았다. 다음 절에서는 동영상 표준 중 하나인 H.264/AVC의 CAVLC와 CABAC에 대해 설명하겠다.

### 1. H.264/AVC CAVLC (Context-based Adaptive Variable Length Coding)

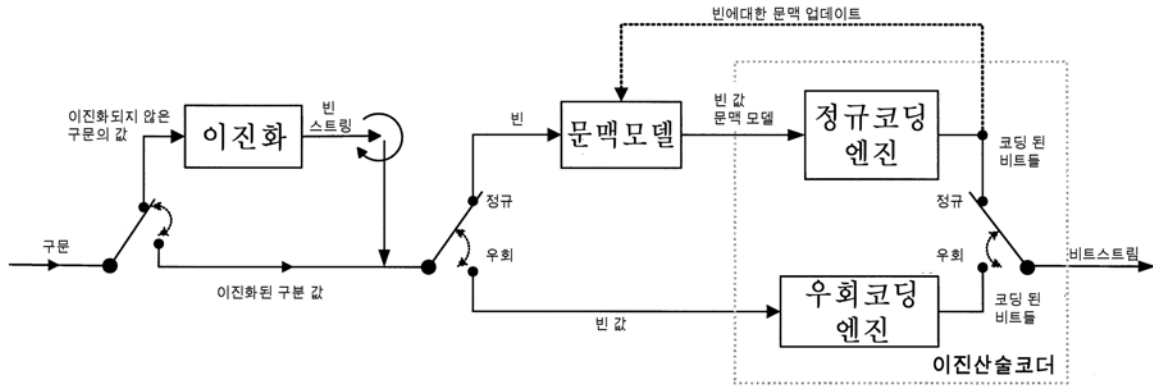
CAVLC는 H.264/AVC 베이스라인 프로파일(baseline profile)

과 익스텐디드 프로파일(extended profile)에서 사용되는 엔트로피 코딩 방법으로, VLC 방법을 기초로 하고 있다. CAVLC는 비트스트림에서 많은 부분을 차지하는 양자화 된 DCT 계수를 부호화 할 때 주로 사용된다. CAVLC의 코딩과정은 여러 단계로 구성되어 있다. 첫 번째 단계는 4×4 블록 단위의 양자화 된 DCT 계수를 지그재스 스캐닝(Zig-Zag scanning)을 통하여 일렬로 나열한다. 스캐닝 된 계수는 앞쪽에는 주파수가 낮은 계수가 분포하고 뒤쪽은 주파수가 높은 계수가 분포하게 된다. 일반적으로 주파수가 낮은 계수의 값은 절대치가 높고, 주파수가 높은 계수는 0 또는 0에 근접한 값을 가지게 된다. 이러한 경우, 일렬로 배열된 계수들은 각 위치에 따라 값에 대한 통계적 특징을 가지고 있으므로, 이러한 점을 이용하여 이후 가변길이 부호화 단계를 수행한다. 두 번째 단계는 일렬로 배열된 양자화 된 DCT 계수를 심볼로 바꾸어 주는 작업을 수행한다. 먼저 낮은 주파수 대역의 양자화 된 DCT 계수 값은 주로 0이 나오므로, 마지막 부분부터 0을 제외하고 양자화 된 DCT 계수에서 1인 값이 연속적으로 나오는 개수(Trailing ones)를 부호화 하고 이에 대한 부호를 부호화 한다. 이 후, 부호화 하지 않은 나머지 계수의 크기(Level)에 대해 가변길이 부호화를 수행한다. 심볼을 가변길이 부호화 하는 과정은 테이블 매핑을 통하여 수행하게 되는데 여기서 테이블을 고정적으로 사용되지 않고 이전의 부호화 된 심볼에 의해 적응적으로 결정된다.

### 2. H.264/AVC CABAC (Context-based Adaptive Variable Length Coding)

CABAC는 H.264/AVC 하이 프로파일(high profile)과 메인 프로파일(main profile)에서 사용되는 엔트로피 부호화 기술로 CAVLC 대비 9%~14% 정도의 높은 압축 성능을 보인다. CABAC이 CAVLC보다 높은 성능을 보이는 이유는 CABAC가 산술코딩을 기반으로 하기 때문이다. 가변길이 코딩보다 산술 코딩이 높은 압축 성능을 내는 이유는 다음과 같다. 가변길이 코딩의 경우 심볼을 코드워드로 변환 과정에서 1:1 대응으로 매핑을 수행하기 때문에 하나의 심볼에 대해서 코드워드의 길이는 최소 길이는 1비트가 된다. 이에 비해, 산술코딩은 부호화 할 일정 영역을 블록이라 할 때 블록 안에 있는 심볼들을 하나의 단위로 보고 산술 부호화를 수행하기 때문에 하나의 심볼에 대해 1비트 보다 작은 단위로 코딩이 가능하여 좀 더 높은 부호화 효율을 낼 수 있다. 하지만 CABAC는 CAVLC에 비해 연산량이 많아 복잡도가 높고 순차적 구조를 가지고 있어 병렬수행이 어려운 문제점이 있다.

<그림 1>은 H.264/AVC CABAC의 블록도이며 과정을 살펴보면 다음과 같다. 영상 부호화 후 나온 구문들의 값은 심볼



〈그림 1〉 CABAC 블록도

이 0과 1만으로 이루어지도록 이진화를 수행한다. 이진화 수행 후 빈 스트링(bin string)이 출력되는데, 각각의 심볼인 0 또는 1을 빈(bin)이라 칭한다. 빈은 이후 진행되는 엔트로피 코딩의 기본단위가 된다. 빈은 문맥 모델을 통하여 확률 값을 예측한다. 여기서, 0과 1 두 심볼 중 낮은 발생확률을 갖는 심볼을 LPS(Least Probable Symbol)라 하며, 높은 발생확률을 갖는 심볼을 MPS(Most Probable Symbol)라 한다. 이후, 문맥 모델 정보와 부호화 할 빈을 이용하여 이진 산술 코딩을 수행하고, 출력은 비트스트림이 된다.

된 기술은 〈표 1〉과 같다.

CAVLC는 맵핑테이블 디자인 및 간략화, 4×4 TU(Transform Unit) 보다 큰 크기에서 양자화 계수의 부호화 방법, 테이블 적응적 업데이트 방법 등의 내용이 제안되었고 CABAC는 주로 문맥 모델링에 관한 내용이 제안되었는데 병렬 문맥 모델 수행을 위한 문맥 설계 또는 문맥 모델링 간략화 등이 있다.

### Ⅲ. HEVC 엔트로피 부호화 기술

현재 HEVC에 채택된 엔트로피 코딩 방법은 고효율 모드와 저복잡도 모드에 따라 두 가지가 있다. 고효율 모드에서는 기존의 H.264/AVC CABAC의 기본 구조를 그대로 채택하였고, 저복잡도 모드에서는 노키아가 제안한 CAVLC가 채택되었다. 이후 두 기술과 관련된 기술들이 제안되었는데, 채택

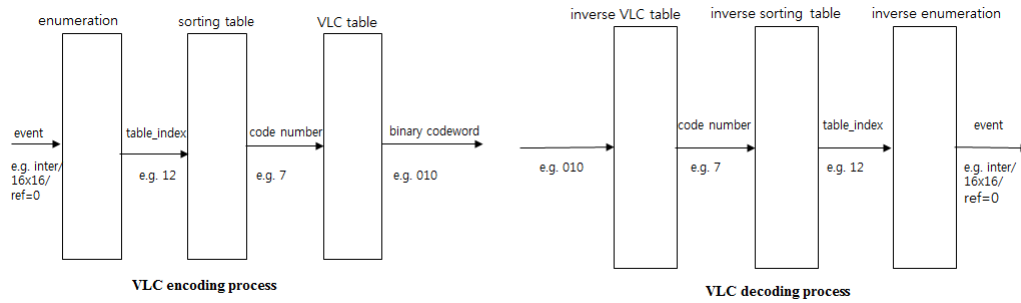
#### 1. HEVC 저복잡도 엔트로피 기술

##### 가. 적응적 테이블 업데이트

현재 HEVC 저복잡도 엔트로피 코딩 기술로 채택된 CAVLC는 VLC기반 엔트로피 코딩방법으로, 구문의 심볼 값을 맵핑테이블에 통하여 코드워드 출력한다. 구문의 특성에 따라 통계적으로 자주 나오는 심볼을 짧은 길이의 코드워드에 맵핑한다. 그리고, 평균 정보량을 줄이기 위하여 발생 확률을 보다 정확하게 예측할 수 있도록 맵핑 테이블을 적응적으로 업데이트 한다. 맵핑 테이블 업데이트 방법은 크게 두 가지가 있다. 하나는 H.264/AVC의 CAVLC의 방법과 같은

〈표 1〉 JCT-VC 5차 회의에서 채택된 엔트로피 코딩 기술

기고서 번호	내용	제안한 기관
JCTVC-E383	CAVLC for 16×16 & 32×32	Qualcomm
JCTVC-E384	CAVLC table size reduction	Qualcomm
JCTVC-E404	LCEC coded block flag coding under residual quadtree	Qualcomm, Microsoft Research
JCTVC-E143	Counter based adaption for LCEC	Qualcomm, USTC, Microsoft
JCTVC-D366	Intra prediction mode coding in LCEC	Qualcomm, Cisco
JCTVC-D370	Inter prediction mode coding in LCEC	Qualcomm
JCTVC-D374	4×4 and 8×8 transform coefficient coding in LCEC	Qualcomm
JCTVC-E227/E489	Context size reduction	Panasonic, TI
JCTVC-E227/E338/E344/E494	significance map coding simplification	Panasonic, Sony, Qualcomm
JCTVC-E324	Modified Context Selection of MVD	MIT
JCTVC-E253	Coding of transform coefficient levels with Golomb-Rice codes	Fraunhofer HHI



〈그림 2〉 파라미터/이벤트 가변길이 부호화 및 복호화 과정

방법으로, 확률 분포에 따라 디자인 된 테이블 중 이전의 부호화 된 구문 발생확률에 따라 가장 적합한 테이블을 고르는 방식이다. 다른 방법은 통계적으로 자주 발생하는 심볼이 다음 부호화 시 짧은 코드워드를 할당 되도록 심볼 간의 순서를 바꾸어 테이블을 정렬하는 방식을 사용한다. 이러한 방법을 파라미터/이벤트 가변길이 코딩이라 한다.

〈그림 2〉는 파라미터/이벤트 가변길이 부호화 및 복호화 과정에 대한 블록도이다. 부호화 과정에서 입력은 부호화 된 구문 값이며, 발생 된 구문 값은 숫자화 하여 테이블인덱스에 매핑된다. 테이블인덱스는 발생 빈도에 따라 코드번호를 할당 받게 되는데 테이블인덱스에 매핑된 코드번호가 작을 경우 현재 구문 값의 발생 빈도가 높다는 것이다. 그러므로 작은 코드번호를 할당 받은 구문 값은 이후 짧은 코드워드를 할당한다. 복호화 방법은 부호화 방법은 역순으로 수행된다.

〈그림 3〉은 파라미터/이벤트 가변길이 코딩의 예이다. 여기서 부호화를 위한 구문 리스트와 이에 대한 우선순위 테이블, 그리고 순위에 따라 짧은 코드워드가 할당되는 VLC 테이블이 존재한다. 과정에 대해 자세히 살펴보면, 부호화 후 구문 리스트 중 결정된 구문 값(Direct\_L0)에 대한 테이블 인덱스는 4이다. 우선순위 테이블에 의해 테이블 인덱스 4에 대한 할당된 코드번호는 4가 된다. 코드 넘버 4는 VLC 테이블을 이용하여 4번째로 짧은 코드워드 0001을 출력한다. 이렇게 출력 된 코드워드는 비트스트림의 일부가 된다. 부호화 이후, 부호화 된 구문 값의 발생확률에 대해서 테이블 업데이트 과정이 이어진다. 발생된 Direct\_L0 구문 값은 발생 빈도가 높아졌기 때문에 보다 짧은 비트를 할당받기 위해 짧은 코드

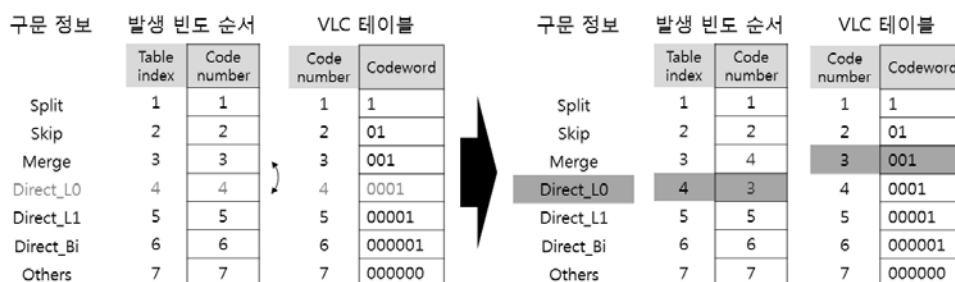
워드를 할당 받은 Merge 구문 값과 코드번호를 바꾼다. 그에 대한 결과로 Direct\_L0 구문 값은 이전의 코드워드 0001보다 짧은 길이를 갖는 코드워드 001에 매핑된 것을 알 수 있다.

### 나. 양자화 된 계수의 부호화 방법

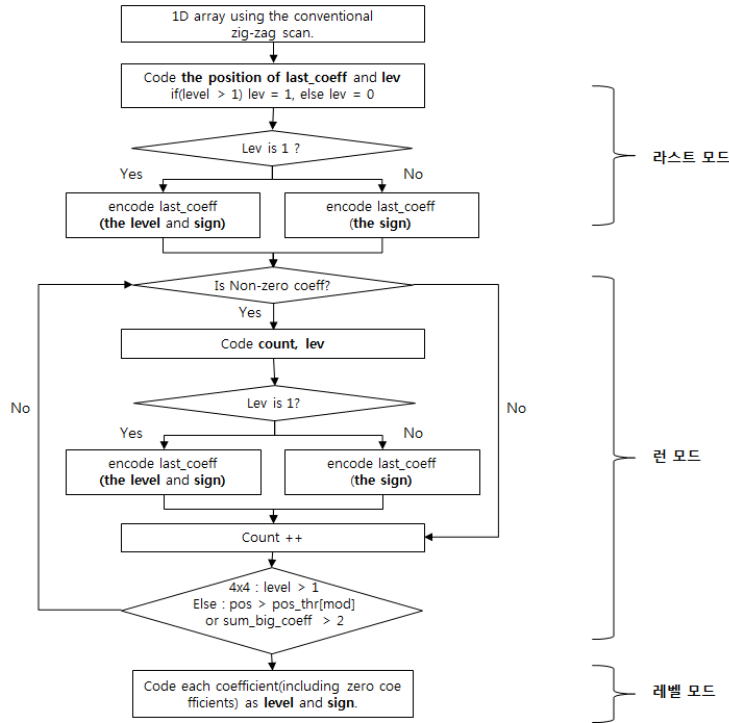
HEVC CAVLC는 양자화 된 DCT 계수를 엔트로피 부호화 하는 과정에 대해 알아본다. 〈그림 4〉는 양자화 된 DCT 계수의 CAVLC 부호화 과정에 대한 흐름도이며 크게 세 가지 단계로 구분할 수 있다. 첫 번째는 TU 내의 양자화 된 DCT 계수들을 고주파에서 저주파로 순으로 스캐닝하여 1차원으로 나열하는 과정, 그리고 처음으로 나오는 0이 아닌 계수의 위치와 크기와 부호를 부호화하는 과정으로 구성된 라스트 모드(last mode)이다. 두 번째는 0의 값을 갖는 계수의 개수와 0이 아닌 값을 갖는 계수를 하나의 단위로 하여 부호화 하는 런 모드(run mode) 단계, 세 번째는 나머지 계수에 대하여 크기 및 부호를 부호화 하는 레벨 모드(level mode)단계로 구분된다. 본 기사에서는 각 단계에 대하여 앞서 말한 것과 같이 첫 번째 단계를 라스트 모드라 하고, 두 번째 단계를 런 모드, 세 번째 단계를 레벨 모드라고 하겠다. 이 후 양자화 된 DCT 계수가 부호화 되는 과정을 좀 더 자세히 알아본다.

#### - 처음 0이 아닌 양자화 계수의 위치 및 계수의 크기 및 부호 부호화 단계 (라스트 모드)

양자화 된 DCT 계수에 대하여 저주파 계수에서 고주파 계수 순으로 나열하여 주파수 특성이 잘 나타나도록 스캐닝 과



〈그림 3〉 파라미터/이벤트 가변길이 코딩 방법



〈그림 4〉 HEVC CAVLC 양자화 된 DCT 계수에 대한 부호화 흐름도

정을 수행한다. 이러한 스캐닝 과정을 거치면 양자화 된 DCT 계수는 1차원으로 나열된다. CAVLC 부호화를 위한 첫 단계는 1차원으로 나열된 배열에서 고투파 성분 쪽부터 저투파 성분 쪽으로 스캐닝을 통하여 처음 0이 아닌 계수를 찾는 것이다. 그리고 찾은 계수의 위치와 계수의 크기가 1인지 아닌지 여부를 last\_pos\_level\_one 구문을 통하여 부호화 한다. 만약 계수의 크기가 1일 경우, 따로 크기 정보를 보내주지 않고 부호만을 부호화한다. 계수의 크기가 1보다 클 경우, 계수의 크기와 부호를 level\_minus2\_and\_sign를 이용하여 함께 부호화 한다. 이 때, 계수의 크기가 1보다 큰지 여부를 이전 구문을 이용하여 알게 되므로 계수의 크기에서 2를 뺀 값을 보내준다. 이 단계에서 앞서 설명한 적응적 테이블 업데이트가 수행된다. 4x4 TU의 경우 발생된 코드번호의 발생 빈도가 높아 순위가 올라가도록 테이블이 업데이트되며, 4x4 보다 큰 TU의 경우 카운팅 되는 수에 따라 코드번호를 부호화하는 VLC 테이블의 종류가 결정된다.

- 연속적으로 0인 계수의 개수와 0이 아닌 계수를 하나의 단위로 하여 부호화 하는 단계 (런 모드)

첫 번째 단계에서 처음으로 0이 아닌 계수의 위치 및 부호에 대한 부호화가 진행된 후에는 런 모드 단계로 넘어가게 된다. 이 단계는 마지막 계수 이후에도 0이 나올 확률이 높기 때문에, 0의 개수와 0이 아닌 계수의 크기를 묶어 부호화하는 단계이다. 런 모드는 임계치가 넘지 않는 범위 내에서 연속적으로 수행되며 일정 조건을 만족할 경우 레벨 모드 단계로 넘어간다. 런 모드를 벗어나는 조건은 블록의 크기, 현재 부호화 할 계수의 위치 정보, 현재까지 0이 아닌 계수 크기의 총 합 등을 고려하여 결정된다.

- 각각의 계수에 대하여 크기 및 부호를 부호화 하는 단계 (레벨 모드)

세 번째 단계는 런 모드 이후에 남은 계수에 대해서 계수의 크기가 0인 경우가 거의 존재하지 않는다고 가정하여 각

〈표 2〉 HEVC CAVLC를 이용한 양자화 된 DCT 계수 구문

모드	구문명	설명
라스트 모드	last_pos_level_one	최초로 0이 아닌 계수의 위치 정보와 계수의 크기가 1보다 큰지 여부를 명시
런 모드	runOfZeros	복호화가 된 위치를 시작으로 이후 0이 아닌 계수가 나올 때까지의 0의 개수를 명시
	levelGreaterThanOneFlag	0이 아닌 계수의 크기가 1인지 여부를 명시
라스트 모드 & 런 모드	level_minus2_and_sign	last mode와 run mode에 0이 아닌 계수가 1보다 클 때, 계수의 크기와 부호를 명시
	sign_flag	last mode와 run mode에 0이 아닌 계수가 1일 때, 계수의 크기와 부호를 명시
레벨 모드	level	계수의 크기를 명시
	sign_flag	계수의 크기가 0보다 클 때, 부호를 명시

각 계수의 크기와 부호를 부호화한다.

〈표 2〉는 양자화된 DCT 계수를 CAVLC로 부/복호화 시 사용되는 구문 정보이다.

#### 다. 화면 내 예측 방향모드에 부호화 방법

H.264/AVC는 양자화 된 계수에 대해서 CAVLC를 사용하였지만, HEVC에서는 발생확률에 대한 통계적 특성을 예측할 수 있는 여러 구문에 대해서 CAVLC를 적용한다. 그 예로 화면 내 예측 모드 구문과 TU의 잔차 신호의 패턴 구문, 화면 간 예측에서 보조 정보에 대한 구문 등이 있다. 그 중 화면 내 예측 모드에 대해서 CAVLC하는 방법에 대해 알아본다. HEVC에서 예측 단위인 PU(prediction unit)의 크기는 기존의 H.264/AVC의 예측 블록에 비해 다양해졌다. 다양한 크기의 PU에 대해 부호화 효율을 높이기 위해 화면 내 예측 방향 모드의 개수가 많아지고, 이에 따라 기존의 H.264/AVC 방법에 비해 보내주어야 할 비트량이 증가한다. 화면 내 예측 시 효율적으로 방향 정보를 보내주기 위해 MPM(Most probable mode)이라는 예측 방법을 사용하여 발생확률이 높은 방향정보를 예측한다. MPM 예측이 실패하게 되면 화면 내 방향모드에 대해서 직접 알려줘야 한다. HEVC에서는 방향 모드의 확률적 특성을 고려하여 CAVLC를 사용한다. 화면 내 예측 방향은 PU 크기에 따라 〈표 3〉과 같이 그 개수가 다르고, 상대적으로 개수가 많은 4×4, 8×8, 16×16, 32×32의 코딩 단위에서만 수행하며, 파라미터/이벤트 가변길이 코딩 방법을 사용한다. 문맥 참조 방법은 적응적 테이블을 사용하고, 적응적 테이블 방법은 이전에 설명했듯이 자주 사용되는 모드는 발생 빈도가 높아져 짧은 코드워드를 할당한다. 64×64모드와 색차성분의 예측 모드는 VLC 방법을 사용하여 부호화한다.

〈표 3〉 화면 내 코딩 단위에 따른 휘도성분 예측 방향의 개수

PU 크기	예측 방향 수
4×4	18
8×8	35
16×16	35
32×32	35
64×64	4

## 2. 고효율 엔트로피 기술 : HEVC CABAC

HEVC CABAC는 〈그림 1〉과 같은 과정으로 부호화를 수행한다. 부호화 수행 전에 초기화 과정이 존재하며, 이후 이진화, 문맥모델, 이진 산술 부호화가 순서대로 수행된다. 먼저 CABAC 초기화에 대해 알아보고 이후 CABAC 부호화 과정의

순서에 따라 부호화 방법을 살펴본다.

### 가. CABAC 초기화

CABAC 초기화 과정은 문맥 초기화와 산술코딩 초기화로 나눌 수 있다. 문맥 초기화는 각 구문이 문맥 모델링을 수행하기 전에 심볼의 발생 확률을 초기화 하는 과정으로 구문의 종류, QP(Quantization Parameter), 슬라이스 타입에 따라 결정되며, 구문 초기화를 위한 파라미터 정보는 테이블화 되어있다. 문맥은 심볼에 대한 예측 발생확률을 저장하는 단위인데, HEVC에서는 MPS의 정보와 LPS의 발생확률을 인덱스화 한 정보(state)를 갖고 있다.

〈표 4〉의 문맥 초기화 수행에 대한 의사 코드를 살펴보면, 입력으로 sliceQP<sub>v</sub>와 m, n을 이용한다. 문맥 초기화 과정을 예를 들어 살펴본다. 문맥 초기화 입력인 m, n, sliceQP<sub>v</sub>의 정보를 알아내야 한다. sliceQP<sub>v</sub>는 슬라이스 헤더에 전송되는 slice\_qp\_delta를 이용하여 계산되며, m, n은 구문의 종류와 슬라이스 타입에 따라 결정되어진다. 이 후 문맥 초기화를 위한 입력인 m, n의 정보를 알아내기 위해 명시된 〈표 5〉를 이용하여 값을 알아낸다. 〈표 5〉는 각 구문마다 어떠한 테이블을 이용하여 구문을 초기화 할 것인지를 저장하고 있다. 그리고 슬라이스 타입에 따라 초기화 파라미터 위치를 결정한다. 예를 위한 구문으로 alf\_cu\_flag의 문맥 초기화 방법을 살펴본다. 이 구문의 역할은 ALF에서 CU(Coding Unit)레벨 on/off 정보이며 슬라이스 헤더에 전송되어진다.

〈표 5〉에서 슬라이스 타입에 따라 문맥 모델을 초기화하기 위해 4번째 열에 사용할 문맥인덱스(ctxid)가 명시되어 있다. alf\_cu\_flag는 table 9-5의 정보를 이용하여 초기화 하며, 여기서 table 9-5는 〈표 6〉과 같다. 슬라이스 타입이 인트라 일 경우, alf\_cu\_flag 구문 초기화 시 사용 될 문맥인덱스는 1, 2, 3이 된다. 〈표 6〉은 결정된 문맥인덱스를 입력으로 하여 초기화 파라미터 m과 n이 결정한다. 초기화에 사용된 문맥 인덱스는 이후 문맥모델링 시 사용된다.

지금까지 문맥 초기화에 대해 알아보았다. 이후, 산술 엔트로피에 대한 초기화에 대해 살펴보면, 산술 코딩을 수행하기 위해 초기 확률 영역은 부호화기와 복호화기 모두 같아야 한다.

〈표 4〉 문맥 초기화에 대한 의사 코드

```
preCtxState = Clip3( 1, 126, ( ( m * Clip3( 0, 51, SliceQPv ) ) >> 4 ) + n )
if ( preCtxState <= 63 ) {
    pStateldx = 63 - preCtxState
    valMPS = 0
} else {
    pStateldx = preCtxState - 64
    valMPS = 1
}
```

〈표 5〉 문맥 초기화를 위한 명시 테이블

	Syntax element	ctxIdxTable	Slice Type		
			I	P	B
slice_header()	alf_cu_flag	Table 9-5	0..2	3,5	6,8
coding_tree()	split_coding_unit_flag	Table 9-6	0..2	3,5	6,8
coding_unit()	skip_flag	Table 9-7		0,2	3,5
	cu_qp_delta	Table 9-8	0,3	4,7	8,11
	pred_type	Table 9-9	0	1,4	5,9
prediction_unit()	prev_intra_luma_pred_flag	Table 9-10	0	1	2
	rem_intra_luma_pred_mode	Table 9-11	0	1	2
	intra_chroma_pred_mode	Table 9-12	0,3	4,7	8,11
	merge_flag	Table 9-13		0,2	3,5
...					

〈표 6〉 alf\_cu\_flag 문맥 초기화를 위한 명시 테이블 (Table 9-5)

Initialisation variables	alf_cu_flag ctxIdx									
	0	1	2	3	4	5	6	7	8	
m	0	0	0	0	0	0	0	0	0	0
n	64	64	64	64	64	64	64	64	64	64

I slice
P slice
B slice

HEVC에서 초기 확률 영역 초기 영역은 0에서 510까지이다. 산술 복호화 시, 확률 영역 안에서의 위치정보를 알아야 복호화가 가능하기 때문에 비트스트림에서 9비트를 읽어 초기 위치를 결정한다.

### 나. 이진화 (Binarization)

엔트로피 부호화 이전의 동영상 부호화 이후 출력으로 나온 구문들은 이진 산술 부호화를 위하여 이진화 되어야 한다. 모든 구문은 이진화를 통해 0과 1의 조합으로 변환되며 이를 빈 스트링이라 한다. 그리고 빈 스트링에 구성된 0 또는 1을 빈이라 칭하며 CABAC의 기본 코딩 단위가 된다. 이진화의 예로 〈표 7〉은 구문을 단항 이진화 과정에서 빈 스트링 출력과 각각의 빈에 대해서 빈인덱스(binIdx)가 결정되는 것을 볼

수 있다. 이진화 종류는 여러 방법이 존재한다. 각 구문의 특성을 고려하여 이에 맞는 이진화 방법을 선택하는데 그 종류로는 고정길이 이진화 방법, 단항 이진화 방법, k-th order exp-golomb 이진화 방법, golomb-rice 이진화 방법 등 여러 방법이 있다. 여기서 golomb-rice 이진화 방법은 H.264/AVC에서 사용되지 않은 새로운 방법으로, 계수의 크기를 보다 효율적으로 이진화하기 위해 제안된 기술이다. 이에 대한 자세한 내용은 참고문헌 [7]에 기술되어 있다.

HEVC에서는 각 구문마다 어떠한 방법을 사용하여 이진화를 수행할 것인지를 〈표 8〉과 같이 명시 되어있다. 〈표 8〉에서 각 구문은 슬라이스 타입에 상관없이 이진화 방법이 동일하다는 것을 알 수 있다. 4번째 열 이후의 내용은 문맥 모델에 대한 정보이다.

〈표 7〉 단항 이진화 방법

Value of syntax element	Bin string						
0 (I_NxN)	0						
1	1	0					
2	1	1	0				
3	1	1	1	0			
4	1	1	1	1	0		
5	1	1	1	1	1	0	
...							
binIdx	0	1	2	3	4	5	

〈표 8〉 각 구문에 대한 이진화 방법 및 문맥 모델 정보

Syntax element	Slice type	Type of binarization	maxBinIdxCtx	ctxIdxTable	ctxIdxOffset
alf_cu_flag	I	FL, cMax = 1	0	Table 9-5	0
	P		0	Table 9-5	3
	B		0	Table 9-5	6
...					
intra_chroma_pred_mode ( IntraPredMode (4 ) )	I	TU, cMax = 3	3	Table 9-12	0
	P		3	Table 9-12	4
	B		3	Table 9-12	8
intra_chroma_pred_mode ( 4 (≠ IntraPredMode (34 ) )	I	TU, cMax = 4	4	Table 9-12	0
	P		4	Table 9-12	4
...					

### 다. 문맥 모델 (Context modeler)

이진화 단계에서 출력인 bin 스트링은 0과 1의 값을 갖는 bin으로 이루어진다. 이러한 bin은 이진 산술 코딩의 기본 단위가 되며 확률을 예측하는 단계인 문맥모델링의 입력이 된다. 문맥 모델링을 통하여 이후 이진 산술 부호화를 수행할 bin의 발생확률 예측 시 예측을 위한 정보로는 현재 bin에 해당되는 구문의 종류와 현재 bin 스트링에서의 위치 등을 이용하여 발생확률을 예측한다.

문맥 모델 과정은 어떠한 문맥을 선택할 것인지를 결정하는 것이다. 문맥 선택은 문맥 위치 정보인 문맥인덱스 결정을 통해 이루어진다. 문맥인덱스를 얻어오기 위해서는 binIdx, maxBinIdxCtx, ctxIdxTable, ctxIdxOffset, ctxIdxInc와 같이 여러 파라미터 정보가 필요하다. 다음은 문맥인덱스 결정을 위한 파라미터에 따라 어떻게 문맥인덱스를 결정하는지 알아본다.

먼저, 〈표 9〉는 문맥을 결정할 때 필요한 명시 테이블이다. 현재 부호화 또는 복호화 할 구문에 대해 ctxIdxTable과 ctxIdxOffset 〈표 8〉 또는 〈표 9〉를 참조하여 얻을 수 있다. 다시 말해, ctxIdxTable과 ctxIdxOffset은 현재 구문이 저장된 테이블 위치와 슬라이스 타입에 따라 분류된 문맥의 위치에 대한 정보이다. 여기서 binIdx는 bin의 위치 정보로, 이진화 후 bin스트링의 위치에 따라 알 수 있다. maxBinIdxCtx는 binIdx

가 너무 많을 경우 이에 대한 상한을 뜻하며, maxBinIdxCtx보다 큰 binIdx의 경우, binIdx는 maxBinIdxCtx 값으로 문맥을 결정한다. maxBinIdxCtx은 〈표 8〉에 4번째 열에 명시되어 있다. 이 세 가지 파라미터로 인해, 〈표 9〉에서 특정 위치의 하나의 칸이 결정된다. 이 칸 안에 여러 문맥이 존재 할 경우 ctxIdxInc 따라 결정된다. ctxIdxInc는 주변 블록의 복호화 된 상태에 따라 결정되며, 이는 각 구문마다 결정하는 방법이 다르다. ctxIdxInc의 기본값은 0이며, 최종 문맥 인덱스는 ctxIdxTable, ctxIdxOffset, ctxIdxInc에 의해 결정 된다.

예를 들어 인트라 슬라이스에서 cu\_qp\_delta 구문의 문맥을 결정해본다. cu\_qp\_delta의 이진화 된 결과인 bin 스트링이 0001이라 하자. 여기서 1에 대한 ctxIdxTable은 Table 9-8이고, 인트라 슬라이스이기 때문에 ctxIdxOffset은 0이 된다.

〈표 9〉에서 na는 이진화 시 복호화 가능한 bin 인덱스가 존재하지 않는 경우이다. cu\_qp\_delta의 bin 스트링인 0001에서 가장 첫 번째 bin을 부호화 하므로, binIdx는 0이다. 이 경우, 해당 위치의 칸 안에 여러 문맥이 존재 하지 않으므로 ctxIdxInc는 0이 된다. 세 가지 파라미터로 인해 cu\_qp\_delta에서 binIdx에 대한 bin의 문맥은 결정되었으며, 결정된 문맥은 Table 9-8의 가장 첫 번째 문맥이다.

위의 내용을 정리하면, 문맥은 구문의 종류와 슬라이스 타입에 따라 나누어지며, 최종적으로 문맥이 결정되기 위해서

〈표 9〉 문맥 결정을 위한 명시 표

Syntax element	ctxIdxTable/ ctxIdxOffset	binIdx					
		0	1	2	3	>=4	
skip_flag	Table 9-7	0	0,1,2	na	na	na	na
		3	0,1,2	na	na	na	na
cu_qp_delta	Table 9-8	0	0	2	3	3	3
		4	0	2	3	3	3
		8	0	2	3	3	3
...							



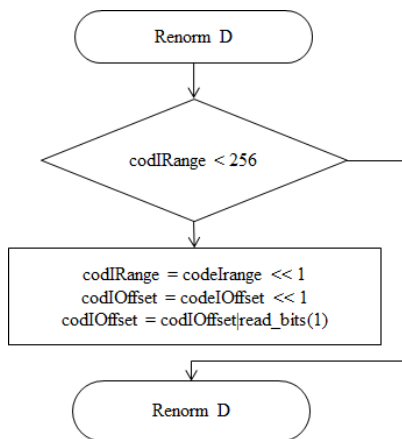
는 부호화 할 bin의 binIdx와 주변 블록의 복호화 결과 등에 따라 결정된다.

양자화 된 계수를 CABAC로 부호화 하는 과정에서, HEVC와 H.264/AVC의 차이점은 변환에 사용된 블록의 크기가 다르다는 것이다. 블록 크기가 다를 경우, 각각의 블록 크기에 대하여 문맥 모델을 설계하여야 한다. 하지만 블록 크기가 커짐으로 인해 그 안의 계수의 양이 많아, 블록 내에 존재하는 각 위치의 계수에 대해서 문맥모델을 설계할 경우 많은 메모리가 필요로 한다. 이러한 점을 고려하여 문맥 모델의 개수를 보다 효과적이게 줄이는 방법에 대해 다수의 기술이 제안되었다. 이에 대한 자세한 내용은 참고문헌 [2]를 참고하길 바란다.

### 라. 이진 산술 코딩 (Binary arithmetic coding)

이진 산술 부호화 할 bin과 문맥모델링을 통하여 얻은 확률 정보를 이용하여 이진 산술 코딩을 수행한다. 이진 산술 코딩은 기존의 산술코딩과 같은 과정을 수행하지만 심볼이 두 가지인 점이 특징이다. 이진 산술 코딩은 문맥 모델에서 예측된 확률로 MPS영역과 LPS영역을 발생확률에 비례하게 나누고 발생한 bin의 값에 해당하는 영역이 다음 영역의 전체가 되는 과정을 반복하여 수행한다. 출력은 마지막 bin이 코딩 된 확률 영역 안의 위치정보가 된다.

실제 컴퓨터의 데이터 저장 방법에서는 무한 실수가 존재하지 않으므로 확률 영역이 일정 크기 이상 줄어들 경우 정밀도 유지를 위하여 재규격화(renormalization)라는 과정을 수행한다. 이는 확률 영역의 위치정보를 부호화 하고 확률 영역 확장을 통하여 영역을 넓혀준다. 복호화 관점에서는 확률 영역이 일정 크기보다 작을 경우 부호화기와 마찬가지로 확률 영역을 확장하고, 위치정보를 비트스트림을 통하여 읽어온다. HEVC는 재규격화를 수행시점은 전체 확률영역이 반보다 작을 경우이며 확률 영역을 2배 확장한다. <그림 5>는 복호화



<그림 5> CABAC 재규격화 흐름도

과정에서 재규격화에 대한 흐름도이다.

### 마. 확률 업데이트 (Probability update)

확률 업데이트는 문맥 모델의 정보를 부호화 또는 복호화 된 구문정보를 이용하여 발생확률을 좀 더 정확하게 예측하기 위한 과정이다. HEVC와 H.264/AVC의 확률 업데이트 과정은 동일하며, <표 10>는 확률 업데이트에 대한 의사코드이다.

다음은 <표 10>에 대해 자세히 살펴본다. binVal은 비트스트림에서 읽어온 1비트 값이며, valMPS는 현재 예측된 문맥 모델을 이용하여 얻어진 MPS 값이다. binVal의 값이 예측된 문맥 모델의 MPS 값을 경우 if문 안의 내용이 수행되고, 아닐 경우 else문 안의 내용이 수행된다. pStateldx는 발생확률을 인덱스화 한 정보이며, 확률 업데이트는 업데이트 테이블을 이용하여 수행한다. 업데이트 테이블은 예측된 문맥모델의 심볼 값과 같은 경우와 그렇지 않은 경우에 대하여 서로 다른 테이블을 사용한다. 전자일 경우에는 확률을 좀 더 높이고 transldxMPS를 이용하여 업데이트가 수행되며, 후자일 경우에는 확률을 낮추도록 transldxLPS를 이용하여 업데이트가 수행된다. 그리고 else문 안에서 pStateldx가 0일 경우는 확률이 0.5인 경우이므로, 발생확률이 높은 심볼 값을 바꾸어준다. 이러한 이유는 실제 pStateldx는 0부터 0.5사이의 확률 값을 표현하기 때문이다.

<표 10> 확률 업데이트에 대한 의사 코드

```

if( binVal == valMPS )
    pStateldx = transldxMPS( pStateldx )
else{
    if( pStateldx == 0 )
        valMPS = 1 - valMPS
    pStateldx = transldxLPS( pStateldx )
}
  
```

## IV. 결론

본 기고는 HEVC 표준화가 진행되고 현재 채택된 기술에 대해서 소개하였다. HEVC 엔트로피 코딩에 채택된 기술은 고효율 모드를 위한 CABAC와 저복잡도 모드를 위한 CAVLC 기술이다. CAVLC는 적응적 테이블 업데이트 기술을 이용하여 이전의 복호화 내용을 바탕으로 적응적으로 테이블을 업데이트 하면서 보다 정확한 코드워드의 선택을 통해 낮은 복잡도 대비 높은 성능을 나타내고 있다. 낮은 복잡성으로 인해 양자화 된 계수뿐만 아닌 다른 여러 구문에 대해서도 CAVLC를 이용하여 부호화 및 복호화를 수행한다. CABAC는 H.264/

AVC와 같은 방법이지만 현재 HEVC의 코딩 구문 및 코딩 단위의 변화로 이에 따라 이진화 방법과 문맥모형을 설계하는 과정에서 기존의 H.264/AVC와는 차이를 보인다. 또한, TU의 크기가 커짐에 따라 양자화 계수의 문맥 모형을 설계하는 과정에서 각 계수의 위치에 따라 문맥을 사용할 경우 많은 메모리를 필요로 하기 때문에 문맥을 효율적으로 줄이는 내용이 포함되어 있다.

앞서 설명한 두 가지 엔트로피 코딩 방법은 각각의 특성에 맞게 기술이 발전하고 있지만, CABAC의 경우 순차적 구조에 대한 문제점과 CAVLC의 경우 좀 더 효율을 높이기 위해 점점 복잡도가 올라간다는 문제점이 있다. 그리고 기술의 통합 및 새로운 하나의 엔트로피 기술을 원하는 많은 의견이 아직 회의에서 대두되고 있다. 현재 HEVC 표준화는 이미 많은 단계에 와 있지만, 통합 엔트로피 코딩 방법 등의 논의가 있어 최종 엔트로피 부호화 톨로 어떠한 프레임 워킹이 채용될지 지켜볼 필요가 있다. 앞으로 HEVC 표준화가 진행되면서 최적의 기술이 표준화로 채택되어 많은 사람들이 기술 혜택을 누리는 날이 도래하길 기대한다.

### 참고문헌

- [1] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Transaction on Circuits and Systems for Video Technology, Vol.13, No.7, pp.620-639, July, 2003.
- [2] Thomas Wiegand, Woo-Jin Han, Benjamin Bross, Jens-Rainer Ohm, and Gary J. Sullivan, "WD3: Working Draft 3 of High-Efficiency Video Coding," JCTVC-E603, Joint Collaborative Team on Video Coding meeting, March, 2011, Geneva, CH.
- [3] Kemal Ugur, Kenneth R. Andersson, and Arild Fuldseth, "Description of video coding technology proposal by Tandberg, Nokia, Ericsson," JCTVC-A119, Joint Collaborative Team on Video Coding meeting, Apr., 2010, Dresden, Germany.
- [4] Marta Karczewicz and Xianglin Wang, "CE5: coefficient coding with LCEC for large blocks," JCTVC-E383, Joint Collaborative Team on Video Coding meeting, March, 2011, Geneva, CH.
- [5] Hisao Sasai, "Simplified Context modeling for Transform Coefficient Coding," JCTVC-D185, Joint Collaborative Team on Video Coding meeting, January, 2011, Daegu, KR.
- [6] Liwei Guo, BinLi, Xianglin Wang, Marta Karczewicz, and Jizheng Xu, "CE5: Counter based adaption for

LCEC," JCTVC-E143, Joint Collaborative Team on Video Coding meeting, March, 2011, Geneva, CH.

- [7] T. Nguyen, M.Winken, D.Marpe, H.Schwarz, and T.Wiegand, "Reduced-complexity entropy coding of transform coefficient levels using a combination of VLC and PIPE", JCTVC-D336, Joint Collaborative Team on Video Coding meeting, January, 2011, Daegu, KR.



유 은 경

2011년 광운대학교 컴퓨터공학과 학사  
2011년~현재 광운대학교 컴퓨터공학과 석사과정  
(관심분야) 영상처리, 비디오 압축, 엔트로피 코딩



심 동 규

1993년 서강대학교 전자공학과 학사  
1995년 서강대학교 전자공학과 석사  
1999년 서강대학교 전자공학과 박사  
1999년~2000년 현대전자(주)  
2000년~2002년 (주)마로비전  
2002년~2005년 University of Washington, Senior Research Engineer  
2005년~현재 광운대학교 컴퓨터공학과 부교수  
(관심분야) 영상처리, 비디오압축, 멀티미디어방송, 패턴 인식