

Versatile Video Codec의 Picture 분할 구조 및 부호화 단위

□ 남다운, 한중기 / 세종대학교

요약

본 고에서는 최근 표준화가 진행되어 2020년에 표준화 완성을 앞두고 있는 VVC의 표준 기술들 중에서 화면 정보를 구성하는 픽처의 부호화 단위 및 분할 구조에 대해 설명한다. 본 고에서 설명되는 내용들은 VTM 6.0을 기준으로 삼는다. 세부적으로는 한 개의 픽처를 분할하는 구조들인 슬라이스, 타일, 브릭, 서브픽처에 대해서 설명한 후, 이 구조들의 내부를 구성하는 CTU 및 CU의 분할 구조에 대해서 설명한다.

1. 서론

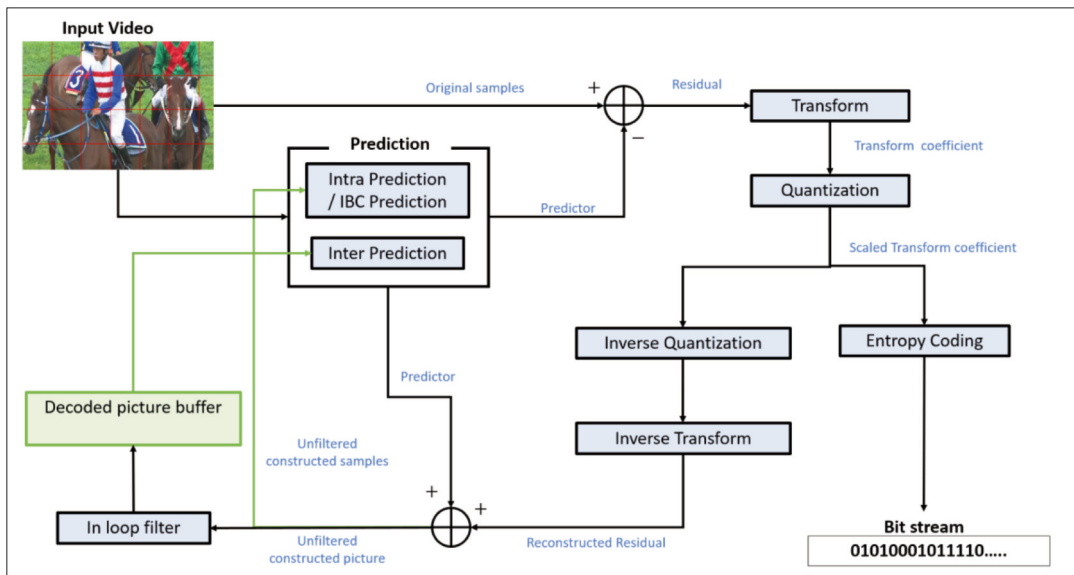
최근 동영상 촬영 장비, 디스플레이 장비와 같은 하드웨어가 발전되고, 동시에 5G 통신망을 통해 실시간으로 고화질 영상의 전송이 가능해졌다. 따라서 4K(UHD)에서 더 나아가 8K 고초화질 영상, 멀티뷰 영상, VR 콘텐츠와 같은 동영상 서비스에 대한 관심이 높아지고 있다. 동시에 고용량 영상 데이

터를 압축할 수 있는 고성능 영상 압축 기술의 필요성이 제기되었다. 고용량 비디오 신호를 압축하는 코덱 기술은 그 시대 산업체의 요청에 따라 계속 발전해왔으며, 이에 따라 다양한 표준 코덱 기술들이 발표되었다. 현재 가장 활발히 쓰이고 있는 동영상 압축 코덱들 중에는 H.264/AVC[1]와 HEVC[2]가 있다. H.264/AVC 기술은 2003년에 ITU-T와 MPEG이 공동으로 표준화한 코덱 기술이고, 높은 압축률로 선명한 영상으로 복호화 할 수 있다. H.264/AVC[1] 개발 후 10년 만인 2013년에 H.264/AVC보다 2배 이상의 부호화 효율을 지원하는 H.265/HEVC(High Efficiency Video Coding)가 개발되었다[2]. 그 후 2018년부터는 ITU-T VCEG (Video Coding Experts Group)와 ISO/IEC MPEG(Moving Picture Experts Group)은 JVET(Joint Video Experts Team)그룹을 만들어, HEVC의 2배의 부호화 효율 개선을 목표로 한

VVC(Versatile Video Coding)의 표준화를 진행하고 있으며, 2020년 7월에 표준화가 완료될 예정이다[3]. 본 고에서는 가장 최신 기술인 VVC에 대해서 설명하며, 특히 VVC에서 영상 신호를 부호화 또는 복호화하기 위한 부호화 단위(Unit)에 대해 자세히 설명하고자 한다.

VVC의 전체적인 부호화 과정은 <그림 1>과 같다. 원본 영상 신호를 입력으로 주면, 화면 내 예측 또는 화면 간 예측을 통해 예측 신호가 생성된다. 원본 신호에서 예측 신호를 빼서 차분 신호를 만들고, 차분 신호를 DCT, DST변환 방법을 이용하여 주파수 영역으로 변환하고, 차분 신호의 변환 계수를 양자화한다. 양자화된 변환 계수를 확률에 따라 코드의 길이를 달리하는 엔트로피 부호화를 통해 최종적으로 비트스트림 형태의 압축된 디지털 신호가 만들어진다. 비트스트림은 데이터 손실을 최소로 하면서 통신망을 통해 전송된다. 수신단에서는

전송 받은 비트스트림 파일의 데이터를 역양자화, 역변환 과정을 거쳐 차분신호로 복호화하고, 부호 화기와 동일한 예측과정을 통해 예측신호를 유도할 수 있다. 그리고 차분신호와 예측 신호를 합하여 복원된 영상 신호를 얻을 수 있다. 이 때 복원 영상은 블록 단위의 부호화에 의한 화질 열화가 발생하고, 이에 의한 주관적 화질의 저하를 막기 위해 인루프-필터링 과정을 거치고 최종적인 복원 영상을 얻을 수 있다. <그림 1>에서 예측 과정은 항상 참조 샘플이 필요하고, 이러한 참조 샘플은 이전에 부호화된 샘플을 다시 복원한 샘플이다. 즉, 다음 부호화 단위에서 부호화될 샘플의 예측을 위해 현재 샘플은 부호화 이후 복호화 과정까지 거쳐야 하며, 이러한 이유로 부호화기 내부에는 복호화기가 포함되어 있다. 본 고에서는 <그림 1>로 설명되는 VVC 부호화기 관련 기술들 중에서 부호화 또는 복호화하는 영상 분할 단위에 대해 자세히 설명하려고 한다.



<그림 1> VVC 부호화기의 구조

VVC의 부호화 과정과 복호화 과정에는 영상을 작은 영역으로 분할(partitioning)하는 과정이 사용된다. 영상 분할에서는 분할된 영역 사이 의존성이 없는 분할 구조도 있고, 분할된 영역 사이 참조가 가능한 의존성이 존재하는 분할 구조도 있다. 영상 안의 슬라이스(slice), 타일(tile), 브릭(brick), 서브픽처(subpicture)는 각각의 다른 슬라이스, 타일, 브릭, 서브픽처들로부터 복호화 의존성이 없으며, 비교적 단순한 분할 구조를 가진다. 슬라이스, 타일, 브릭, 서브픽처는 각각의 특정 목적을 위해서 정의되었으며, 기본적으로 여러 개의 CTU(Coding Tree Unit) 유닛으로 구성된다. CTU는 이웃한 CTU를 참조할 수 있으며(의존성이 있으며), CTU는 QT(Quad Tree) 또는 MTT(Multi-Type Tree)구조로 분할되어 CU라는 여러 개의 하위 블록이 만들어진다. 본 고에서는 영상을 구성하는 슬라이스, 타일, 브릭, 서브픽처의 의미와 기능을 설명하고, 영상이 CU로 분할되는 전체적인 과정을 설명한다.

본 고의 구성은 다음과 같다. 본 고의 II 장에서는 VVC 참조 소프트웨어(Reference Software)의 최신 버전인 VTM 6.0[4]를 기준으로, 슬라이스, 타일, 브릭, 서브픽처의 구조를 설명하고, III장에서는 부호화 블록인 CTU, CU의 분할 구조를 설명한다. IV장에서는 하드웨어 상의 디코더 과정을 고려한 VPDU 분할에 대해 설명한다. 마지막으로 V장에서 간단한 결론을 맺는다.

II. 슬라이스, 타일, 브릭, 서브픽처의 분할 구조

본 장에서 설명하는 슬라이스, 타일, 브릭, 서브

픽처는 사용자에 의해 분할 종류, 분할 크기, 분할 개수가 정해질 수 있다[4].

1. 슬라이스(Slice)

영상이 부호화 과정을 통해 비트스트림으로 변환되어 전송될 때, 전송 채널을 지나 수신단까지 전달된다. 하지만 전송 채널에 채널 에러가 심하게 존재할 경우, 전송된 데이터의 일부 또는 전체가 손상될 수 있다. 이런 환경에서 한 픽처 전체를 압축해서 하나의 패킷에 담아 전송한다면, 해당 패킷의 일부만 손상되어도 한 픽처 전체의 화소값들을 복호화할 수 없게 된다. 저화질 영상은 통째로 비트스트림으로 전송하는 경우도 있지만, 고화질 영상의 경우 채널 에러를 고려해 여러 개의 패킷으로 나눠 네트워크 상으로 보낼 필요가 있다. 영상을 여러 개의 패킷으로 쪼개 전송할 때, 만약 열악한 네트워크 상태로 인해 하나의 패킷이 손실되어도 손실된 패킷을 제외한 영상의 나머지 부분은 복원할 수 있다. 네트워크 상으로 보내지는 패킷은 NAL 유닛 기반으로 만들어지며, 일반적으로 하나의 NAL 유닛 안에 포함되는 영상 데이터는 하나의 슬라이스(slice)에 포함된 데이터이다. 즉, <그림 2>와 같이 영상(picture)을 더 작은 영역인 슬라이스로 분할하고, 슬라이스 데이터를 패킷으로 캡슐화하여 수신단으로 전송할 수 있다. 현재 슬라이스를 복호화 할 때, 주변 슬라이스 데이터의 손실이 있을지라도 그와 관계없이 독립적으로 복원될 수 있어야 하기 때문에, 슬라이스 사이에는 복호화 의존성은 존재하지 않는다. 하나의 정지 영상은 하나 이상의 슬라이스로 분할될 수 있다. 하나의 슬라이스(또는 타일)는 하나 이상의 브릭(brick)으로 구성된다. 슬라이스는 레스터-스캔 슬라이스(raster-scan slice)와 직사

각형 슬라이스(Rectangular slice)로 2가지 종류가 있다. 레스터-스캔 슬라이스는 레스터-스캔 순서로 연속된 타일(Tile)들의 그룹으로 구성된다. 직사각형 슬라이스는 이름의 의미 그대로, 영상안의 직사각형 영역들로 분할된 슬라이스를 의미한다. <그림 2>의 입력영상은 Class B의 MarketPlace 시퀀스의 첫번째 프레임이고, 슬라이스 종류에 따른 분할 형태의 차이를 확인할 수 있다.

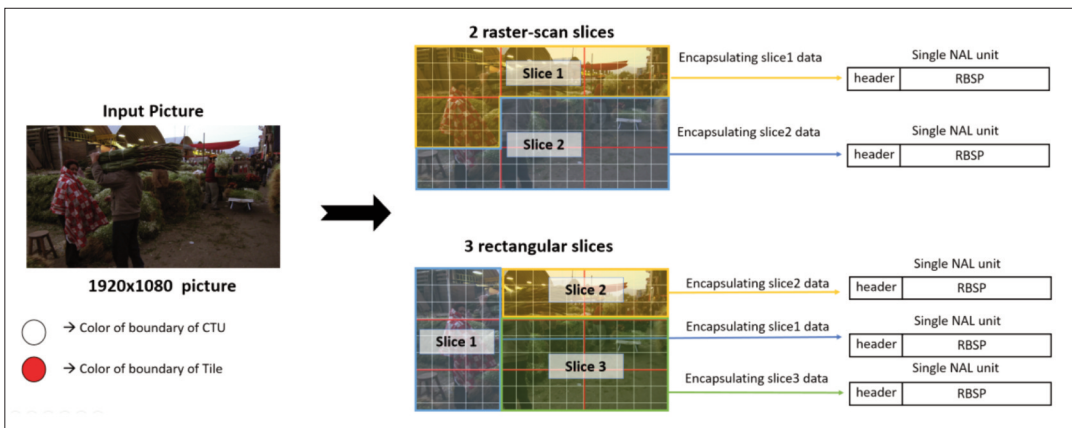
2. 타일(Tile)

고해상도(Ultra High Definition), 고비트심도(High Bit Depth), 고프레임률(High Frame Rate) 영상과 같은 고용량의 동영상을 끊어짐 없이 실시간 복호화하기 위해서는 병렬처리가 필수적이다. 타일(Tile)은 영상 안에 직사각형으로 분할된 영역이고, 타일 단위로 부호화와 복호화 과정의 병렬 처리가 가능하다. 즉, 타일 단위로 공간적 예측과 엔트로피 코딩 과정이 독립적으로 적용될 수 있다. 타일분할을 통해 병렬처리에 의한 복호화 과정으로 고속화를 할 수 있지만, 압축 효율이 줄어들고 불필요한 타일

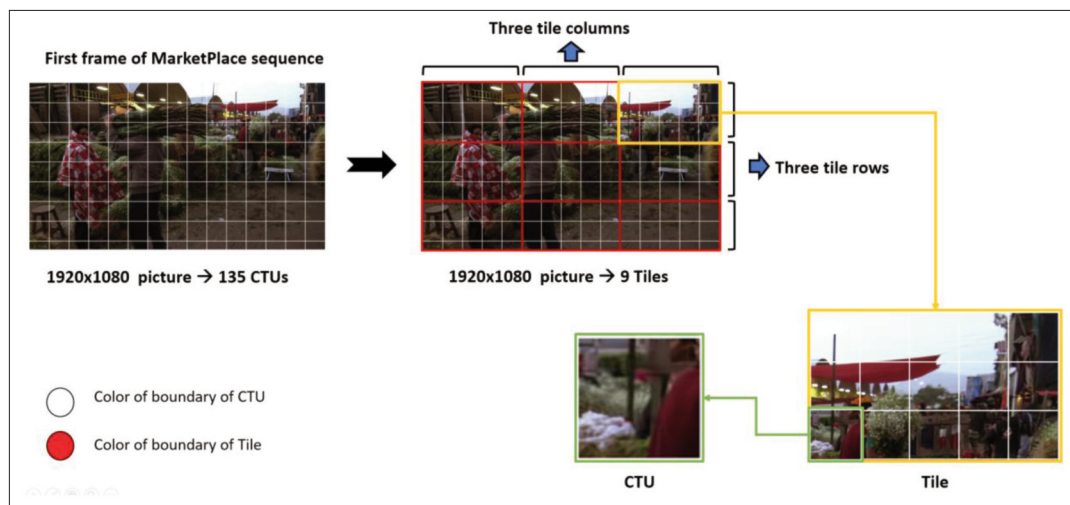
경계에 열화(tiling artifact)가 발생할 수 있다.

VVC 타일의 분할 구조를 보면, 현재 영상에서 타일 구조를 사용하면, 영상 안에는 하나 이상의 타일 행과 하나 이상의 타일 열로 나눌 수 있다. 즉, 하나의 영상(또는 slice)은 타일 열(Tile columns)의 개수×타일 행(Tile row)의 개수로 표현되는 타일들로 나눌 수 있다. 이때 uniform_tile_spacing_flag라는 선택스 요소에 의해, 각각의 타일 행의 높이와 타일 열의 너비가 균일하게 분할될 수 있고, 직접 각각의 타일 행의 높이와 타일 열의 너비를 지정해 줄 수도 있다. 그리고 타일은 하나 이상의 브릭(brick)을 포함하며, 동시에 브릭은 정수 개의 CTU를 포함한다.

추가적으로, 이런 타일의 특성을 이용하여 HEVC 표준화 이후 MCTS(Motion Constraint Tile Set)기술이 등장하였다[5]. MCTS는 타일 간의 공간적, 시간적 독립성을 가질 수 있도록 하는 부호화기(encoder)쪽 기술로서, 현재 타일이 속한 MCTS에 포함되지 않는 타일들의 정보를 참조할 수 없게 된다. 영상을 MCTS로 분할하여 부호화하면, MCTS의 독립적으로 전송, 복호화가 가능해진다. MCTS는



<그림 2> Slice의 종류



〈그림 3〉 Tile의 구조

MPEG OMAF(Omnidirectional Media Format) 표준을 기반으로 한 사용자 시점 기반 VR 어플리케이션에서 중요한 의미를 가진다. 실시간으로 360도 VR 영상을 스트리밍할 때, 전체 영상을 전송하고 복호화하기에는 대역폭의 한계가 있기 때문에, 실제 사용자 시점에 해당하는 영역만 추출하여 더 높은 해상도로 처리하는 것이 효율적이다. 이 때, MCTS기술을 이용하면 사용자 시점 뷰포트(viewport)에 해당하는 타일 세트의 독립적인 전송 및 처리가 가능해진다. 이는 II-4에서 설명할 VVC의 서브픽처(subpicture) 개념과 연관이 있다.

3. 브릭(Brick)

브릭은 2019년 3월 제네바 JVET 미팅에서 논의된 분할 구조[6]로, VVC의 분할 구조의 유연성을 높이기 위한 분할 구조로 제안되었다. Brick은 HEVC에서는 존재하지 않았던 개념으로, HEVC에서는 슬라이스와 타일의 구성요소가 CTU 뿐이었다. 반면

VVC에서는 슬라이스와 타일을 구성하는 요소가 CTU 뿐만 아니라 브릭(Brick)도 존재한다. 브릭은 공간적 예측 과정에서 이웃 브릭에 대한 의존성이 없으며, 브릭 안의 첫번째 CTU에 대해 CABAC 초기화 과정이 존재한다. 브릭은 타일과 유사한 복호화 독립성을 가지며, VVC는 브릭과 타일을 이용해 보다 더 유연한 타일링(tiling)을 지원할 수 있다.

영상 안의 분할 구조로 보면, 영상 안의 각각의 슬라이스와 타일은 하나 이상의 브릭으로 구성되며, 브릭은 하나 이상의 CTU로 구성되는 직사각형 영역이다. 이 때 주의할 점은 브릭은 타일을 수평으로 분할한 것으로, 수평으로 분할되지 않는 타일은 그 자체로 하나의 브릭으로 간주된다.

타일과 브릭과 CTU로 이어지는 계층적 분할 구조에서 스캔(부호화) 순서는 다음과 같다. 먼저 한 영상(picture) 내부의 복수개 타일들은 레스터 스캔(raster-scan) 순서로 부호화되고, 하나의 타일 안의 브릭들은 레스터 스캔 순서로 부호화된다. 하나의 브릭 안의 CTU들도 마찬가지로 레스터 스캔 순

서로 부호화된다[7].

4. 서브픽처(Subpicture)

서브픽처(Subpicture) 개념은 2019년 1월에 모로코 마라케쉬에서 개최되었던 JVET 회의에서 본격적인 논의가 시작되고, 2019년 7월 스웨덴 예테보리에서 개최되었던 JVET 회의에서 MCTS 기능을 지원하기 위한 의도로 채택되었다[8]. 직사각형 타일 그룹을 논의하면서 더 유연한 CVS의 제공을 위한 새로운 분할 구조로 서브픽처를 제안하였다.

영상을 서브픽처로 분할하여 독립적인 부호화가 가능하며, 과거 표준화 기술인 HEVC의 MCTS와 유사하게 뷰포트 기반 360도 VR영상 스트리밍에서 중요한 의미를 가진다. HEVC 부호화기에서 MCTS로 분할하여 부호화하고, 비트스트림을 추출(extraction) 및 병합(merging)하여 타일 단위의 복호화와 렌더링 과정이 가능했다. 이와 유사한 기능을 위해 VVC 6에서 서브픽처 단위로 독립적으로 부호화될 수 있으며, 서브픽처 단위로 비트스트림 추출 및 병합하여 서브픽처 단위로 복호화될 수 있다. 서브픽처 구조는 MCTS보다 더 유연한 분할 구조가 가능하며, 율-왜곡 효율 관점에서 더 효율적이다. 각각의 서브픽처의 위치와 크기 등의 정보는 SPS에서 시그널된다. 필요에 따라 'subpic_treated_as_pic_flag'라는 SPS 선택스 요소에 의해 하나의 서브 픽처는 복호화 과정(인-루프 필터 제외) 안에서 하나의 영상(picture)인 것처럼 다뤄질 수 있다. 영상 내 서브픽처의 분할 구조를 보면, 영상은 하나 이상의 서브픽처로 분할될 수 있고, 하나의 서브 픽처는 하나 이상의 직사각형 슬라이스(Rectangular mode slice)로 구성되는 직사각형 영역이다[3].

III. 부호화 블록 구조

슬라이스(slice), 타일(tile), 브릭(brick), 서브픽처(Subpicture)는 사용자에 의해 크기 및 개수를 결정할 수 있으며, 어플리케이션의 특성에 따라 다양하게 선택할 수 있다. 이 때 슬라이스, 타일, 브릭, 서브픽처는 다수의 CTU들로 구성되어 있다. 그리고 각 CTU들마다 재귀적 분할 구조를 통해 여러 CU들이 생성된다. 그리고 CU단위로 예측, 변환, 양자화, 엔트로피 부호화, 복호화까지의 연속된 과정이 적용되고, 복원된 CU가 저장된다. 그 복원된 CU 정보를 기반으로 다음 CU의 부호화 과정이 이어서 진행된다. CTU의 CU가 모두 부호화되면 레스터 스캔 순서에 따라 다음의 CTU안의 CU가 부호화된다.

1. CTU와 CU

영상은 가장 먼저 균일한 CTU들로 분할된다. HEVC와 VVC에서 CTU는 CU로의 분할 트리 구조에서 가장 상위 노드가 되며, CU의 최대 크기는 CTU의 크기와 같다. CTU와 CU에서 주의할 점으로 YUV 색상 형식의 컬러 영상의 경우, CTU는 한 개의 휘도 성분(luma)의 CTB(Coding Tree Block)와 두 개의 색차 성분(chroma)의 CTB(Coding Tree Block)를 포함하고, CU는 한 개의 휘도 CB(Coding block)와 두 개의 색차 부호화 CB를 모두 포함하는 개념이다.

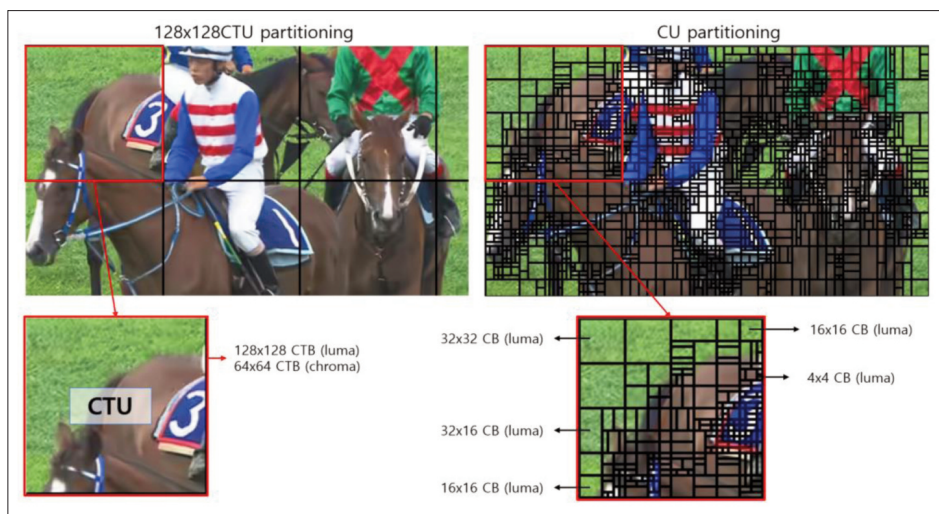
〈그림 4〉는 VTM6.0로 416x240 저해상도 영상을 부호화 블록 CU로 분할된 모습을 보여준다. 왼쪽 그림은 영상을 휘도 성분 기준 128x128 크기로 CTU 분할을 검은색 선으로 표현하였다. 오른쪽 그림은 모든 CTU에 대해 CU로 분할된 모습을 검은

색 선으로 표현하였다. <그림 4>에서 보이는 것과 같이 CTU의 크기는 균일하지만, CTU 내부에 분할된 CU들은 최소 4x4부터 32x32(실제 허용 가능한 최대 CU 사이즈는 128x128)까지 다양한 크기를 가지고, 모양도 정사각형 뿐만 아니라 다양한 모양의 직사각형 역시 허용되었음을 확인할 수 있다.

<그림 4>의 왼쪽 그림은 YUV 4:2:0 형식인 Class D의 RaceHorses 시퀀스의 첫번째 프레임을 CTU로 분할한 것이다. 하나의 CTU는 해당 위치의 128x128 휘도 블록과 2개의 64x64 색차 블록으로 구성된다. CTU는 HEVC부터 등장한 개념으로 VVC와 비교하여 동일한 기능을 가진다. 하지만 HEVC에서는 CTU 크기로 최대 64x64의 크기까지 허용되었지만, VVC에서는 128x128 크기가 허용된다. 이에 따라 VVC는 HEVC보다 CU에 허용된 크기의 범위가 넓어졌다. 또한 HEVC에 비해 다양한 분할 모양의 CU가 가능해졌다. 이로 인해 부호화 성능이 증가였지만, 부호화의 복잡도가 크게 증가하였다.

2. 부호화 블록(CU) 분할 구조

CTU에서 CU로의 분할 트리의 종류로는 이진 트리(BT, Binary Tree), 삼진 트리(TT, Ternary Tree), 사진 트리(QT, Quad Tree)가 있다. 먼저 QT는 하나의 노드로 4개의 하위 노드를 생성하는 트리 구조로써, 현재 블록에 QT 구조를 적용하면 4개의 같은 크기의 블록으로 분할된다. 이 분할 구조는 HEVC의 참조 소프트웨어 모델인 HM에서 CTU에서 CU로 분할할 때 사용한 방법으로, HM에서는 오직 QT 구조만을 사용하여 분할하며, 최적의 분할 구조가 나타날 때까지 재귀적으로 블록을 같은 크기로 사등분한다. 두 번째로 BT란 하나의 노드에서 2개의 하위 노드를 생성하는 트리로, BT를 통해 한 블록을 같은 크기의 두 개의 하위블록으로 분할할 수 있으며, 분할 방향은 수평 또는 수직이 된다. HM과 VTM의 중간 과정인 JEM(Joint Exploration Model)에서는 QT와 BT를 모두 사용하는 QTBT 방식을 사용해 CU로 분할하였다. 이 때, 이전 JEM,

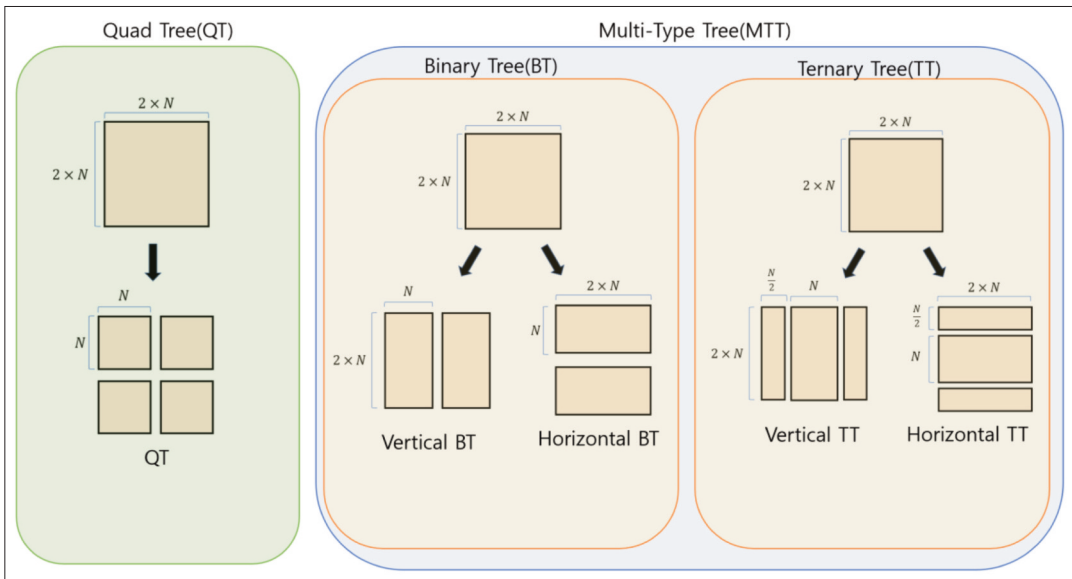


<그림 4> CTU 분할과 CU 분할

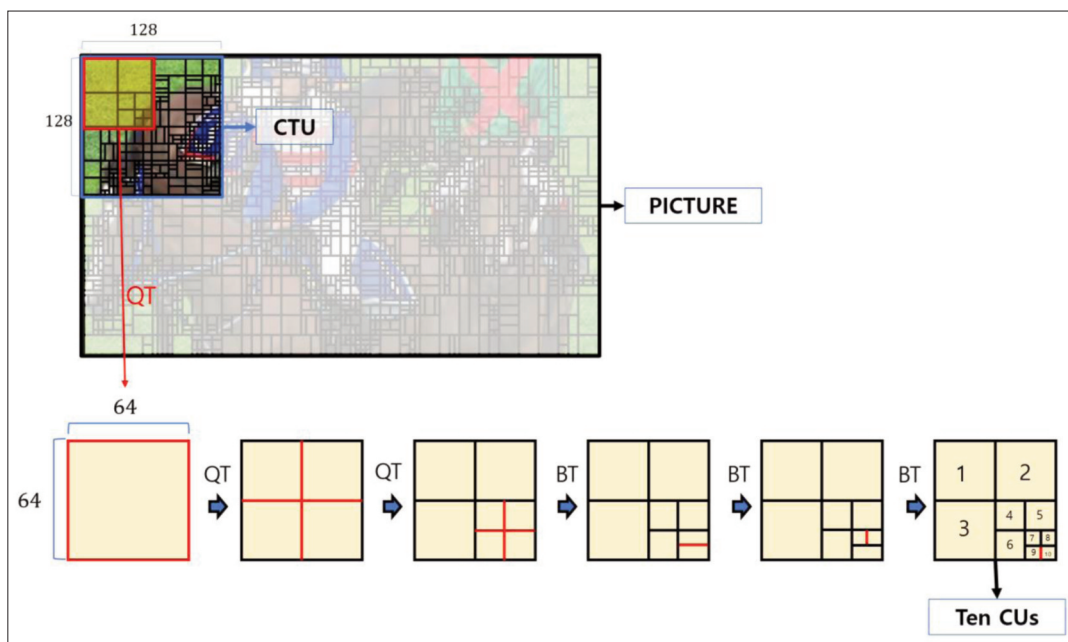
HM과 비교하여 5%의 BD-rate 개선이 되었다. 반면 부호화 복잡도가 크게 증가하였다[10][11]. TT란 하나의 노드에서 세 개의 하위 노드를 생성하는 트리이지만, 주의할 점은 블록 분할에 적용되는 TT는 같은 크기의 블록으로 삼등분되는 것이 아니라, 가로 또는 세로의 길이가 1:2:1 길이 비율로 삼등분되는 것이다. BT와 마찬가지로 분할방향은 수평 또는 수직이 될 수 있다. 현재 VVC는 QTBT 방식에 TT를 추가한 기술을 사용하여 CU분할을 하고 있다[11][12]. 현재 VVC 6.0 드래프트 문서에서는 BT와 TT를 묶어 MTT(Multi-Type Tree)로 구분하였다. 이 QT와 MTT를 사용하는 분할 구조는 이전 JEM 버전에 비교하여 3~4% BD-rate 개선이 되었다. 반면 부호화 복잡도는 2배 가까이 증가하였다. <그림 5>는 QT, BT, TT로 분할 시 하위 블록의 모양과 크기를 나타낸 것이다.

3. CU 분할의 전체 과정

CU는 최소 4x4에서 최대 128x128까지의 다양한 크기와 형태를 가질 수 있다. CTU로부터 CU까지의 전체적인 분할 과정은 다음과 같다. 가장 먼저 CTU의 QT로 분할된다. QT는 허용 가능한 크기(최소 16x16, 최대 128x128) 안에서 재귀적으로 분할될 수 있다. QT 구조를 통한 하위 노드는 MTT 분할이 가능한 크기 조건과 깊이 조건을 만족하면 MTT로도 분할이 가능하다. 그리고 한 번 MTT 분할이 적용된 노드는 더 이상 QT 분할이 적용되지 않고, MTT 분할이 재귀적으로 적용될 수 있다. VTM 6.0에서는 허용되는 크기와 깊이 조건 내에 가장 RD-cost 값이 작은 분할 구조가 선택되고 동시에 CU가 결정되었다. <그림 6>을 보면, 일반적으로 저주파 영역에는 큰 CU로, 고주파 영역은 작은 CU로 선택된다는 것을 확인할 수 있다.



<그림 5> Quad-Tree(QT)와 Multi-Type-Tree(MTT)로의 블록 분할



〈그림 6〉 CU까지 분할 과정

〈그림 6〉은 128x128 CTU에서 QT 분할로 생성된 4개의 64x64 블록 중 첫 번째 64x64 블록에 대해 CU로의 분할되는 과정을 보여준다. QT는 허용되는 최소 QT 크기까지 분할된 후, BT분할이 재귀적으로 적용되어 최종적으로 10개의 CU가 생성된다. 이 때, CU의 스캔 순서는 각 분할 구조 안의 레스터 스캔 순서를 따른다.

4. 변환 블록(TU) 분할

HEVC에서는 CTU를 CU로 분할하고, CU에서 다시 PU(Prediction Unit)와 TU(Transform Unit)로 분할하였다. CU는 부호화 및 복호화 과정이 적용되는 기본 단위이며, PU는 예측 블록이 생성되는 기본단위가 되고, TU 단위로 차분 신호의 변환, 양자화, 엔트로피 코딩까지의 과정의 기본단위가 된

다. HEVC에서와 달리, VVC에서는 CU와 PU, TU는 동일한 블록이다. 하지만 예외적으로 CU가 최대 TU의 크기를 넘어서는 경우에는 CU와 TU는 동일하지 않다. 휘도 신호에서 TU가 될 수 있는 가능한 크기의 범위는 4x4 ~ 64x64이다. 그리고 현재 영상에서 사용되는 최대 TU의 크기는 이 후 IV장에서 설명할 VPDU의 크기가 된다.

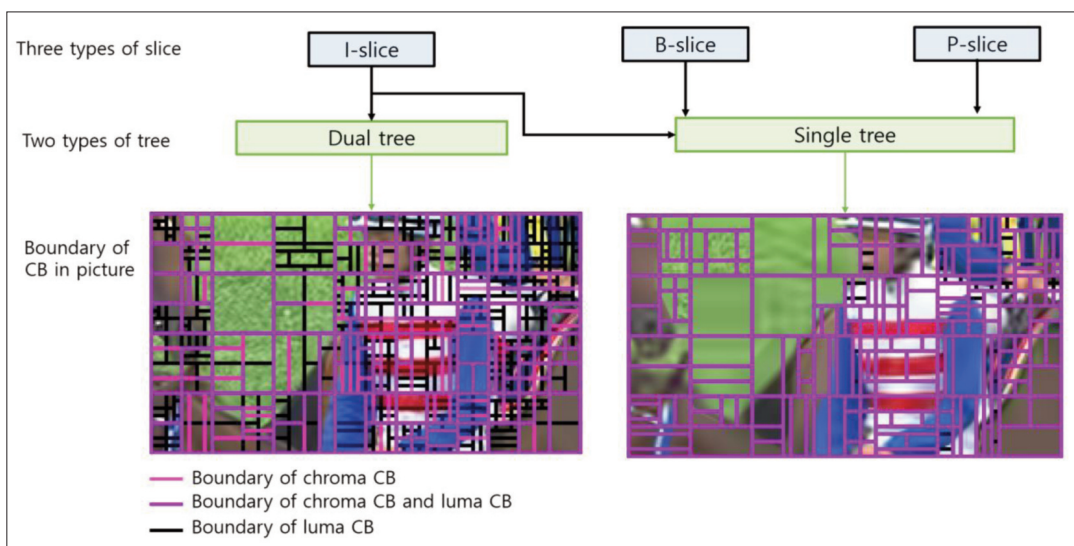
5. 슬라이스 종류에 따른 분할 트리 구조의 종류

앞서 II-1절에서 설명한 슬라이스는 3가지 종류가 존재하고, 슬라이스 종류에 따라 해당 슬라이스 안에 포함되는 CU의 예측 모드가 달라진다. 슬라이스의 종류로는 'I 슬라이스', 'P 슬라이스', 'B 슬라이스'로 구분되고, 'I 슬라이스' 안의 CU는 화면

간 예측과 화면 간 예측에 관련된 톨들을 사용할 수 없다. ‘P 슬라이스’와 ‘B 슬라이스’ 안의 CU는 화면 간 예측을 사용할 수 있으며, P 슬라이스는 화면 간 예측 시 한 개의 참조 영상 리스트(reference picture list)에서 참조 영상을 선택할 수 있고, B 슬라이스에서 두 개의 참조 영상 리스트로부터 참조 영상을 선택하여 화면 간 예측을 할 수 있다. 이 때 슬라이스의 종류에 따라 CTU에서 CU까지의 분할 구조가 다를 수 있다.

일반적으로 부호화하려는 영상은 YUV 컬러 영상으로 휘도 신호와 2개의 색차 신호로 분리된다. 하지만, 휘도 신호와 색차 신호는 유사하지만 다른 특성을 가지고 있기 때문에 실제로 최적의 분할 구조도 다를 수 있다. 하지만, 휘도 신호와 색차 신호에서 개별적으로 분할 구조를 결정하면 복잡도가 크게 증가한다. 이 때, 현재 슬라이스가 I 슬라이스라면 휘도 신호의 분할과 색차 신호의 분할이 독립적인 듀얼 트리(Dual tree) 구조가 적용될 수 있다.

P 또는 B 슬라이스에서는 휘도 성분과 색차 성분에서 동일한 분할 구조를 가지는 싱글 트리(Single tree)가 적용된다. 이 때 I 슬라이스는 듀얼 트리의 사용여부를 나타내는 플래그를 통해 듀얼 트리와 싱글 트리 중 하나의 방식으로 선택된 후 그에 따라 부호화될 수 있다. 반면, HM에서는 듀얼 트리 구조가 적용되지 않았고, 휘도 신호와 색차 신호에 대해 동일한 블록 분할 구조가 사용되었다. 따라서 HM에 비교하여 VTM은 I 슬라이스에서 듀얼 트리를 적용하는 경우 색차 신호의 BD-rate가 개선되지만 복잡도가 크게 증가하게 되었다. <그림 7>은 색차 신호의 CB의 경계선과 휘도 신호의 CB의 경계선을 나타낸 그림이다. <그림 7>의 왼쪽 그림은 I 슬라이스인 영상으로 듀얼 트리 구조로 분할하였고 색차 신호와 휘도 신호의 분할 구조가 다르다는 것을 확인할 수 있다. 오른쪽 그림은 P 슬라이스인 영상으로 싱글 트리 구조로 블록이 분할되었고 색차 신호와 휘도 신호의 블록 분할은 색차 신호의 블록이 최



<그림 7> Dual Tree와 Single Tree의 색차 성분 CB와 휘도 성분 CB의 분할

소 크기 제한을 벗어나지 않는 경우, 거의 동일한 구조임을 보여준다.

IV. 하드웨어 파이프라이닝과 VPDU

복호화기의 하드웨어 관점에서, 복호화 과정을 여러 파이프 라인 단계로 나눌 수 있다. 이를 통해 여러 파이프 라인 단계의 명령을 동시에 처리하는 파이프라이닝을 할 수 있다. 이 때 각 파이프라인 단계에서 입력되거나 출력되는 데이터 단위를 VPDU(Virtual Pipeline Data Unit)라 한다[13][14]. VPDU는 CTU를 넘을 수 없는 블록으로, VPDU는 CU를 완전히 포함하거나 CU안에 완전히 포함된다. VPDU는 전체 CTU의 액세스없이 복호화 처리 과정에 사용되는 데이터 유닛이다. 영상을 VPDU로 분할하여 여러 파이프라인 단계에 VPDU단위로 입력을 주면, 연속된 VPDU를 동시에 다른 파이프라인 단계에서 처리할 수 있다. 현재 파이프라인 단계에서 출력된 데이터가 다음 단계의 입력으로 이어질 때, 기다리는 시간이 생기지 않도록 나뉜 각 파이프라인 단계는 처리 시간이 비슷해야 하고, 각 파이프라인에 필요한 버퍼의 크기는 비슷해야 한다. 이 때, 파이프라인에서 처리될 수 있는 데이터

의 크기는 버퍼 사이즈에 비례한다. 따라서 VPDU는 대부분 파이프라인 단계의 버퍼 크기에 비례하고, HEVC에서는 최대 TU 크기(휘도 성분 32x32 블록, 색차 성분 16x16)로 VPDU가 설정되었으며, VVC에서도 최대 TU 크기(휘도 성분 64x64 블록, 색차 성분 32x32)로 VPDU가 설정되었다. 일반적으로 하드웨어 관점에서 파이프라인 단계를 더욱 작게 분할하면 파이프라이닝에 의한 처리 속도가 높아질 수 있으며 메모리 절약을 위해 VPDU 크기는 작게 유지하는 것이 중요하다. VVC에서는 VPDU 크기를 고려하면서, 큰 사이즈 블록의 MTT 분할과 예측 과정의 참조에 대해 제한이 생겼다.

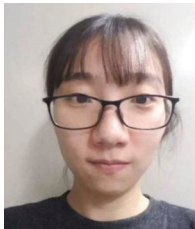
V. 결론

본 고에서는 2019년 9월 현재까지 표준화가 진행되면서 만들어진 VTM 6.0를 기준으로 화면 정보를 구성하는 픽처의 부호화 단위 및 분할 구조에 대해 설명했다. 또한, 표준화가 완성되어 산업체에서 다양하게 활용되고 있는 HEVC 기술과 세부 내용을 비교하고 차이점을 설명하였다. 본 기고문을 통해 독자들이 VVC 관련 전문 지식을 습득하고 관련 분야의 기술 발전에 도움이 되기를 바란다.

참고 문헌

- [1] ITU-T, "Recommendation and International Standard of Joint Video Specification," ITU-T Recommendation H.264/ISO/IEC 14496-10 AVC, October, 2004.
- [2] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, Y. K. Wang and T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Last Call)," The Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-L1003_v34, January, 2013
- [3] Benjamin Bross , Jianle Chen and Shan Liu , "Versatile Video Coding (Draft 6)," JVET-O2001, Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3-12 July 2019
- [4] Jianle Chen, Yan Ye and Seung Hwan Kim, "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)," JVET-O2002, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3-12 July 2019
- [5] R. Skupin, Y. Sanchez, K. Suehring, T. Schierl, E. Ryu, and J. Son, "Temporal MCTS Coding Constraints Implementation," 120th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG 120/m41626, 2017.
- [6] Gary Sullivan and Jens-Rainer Ohm , "Meeting Report of the 14th Meeting of the Joint Video Experts Team (JVET), Geneva, CH, 1927 March 2019," JVET-N1000, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19-27 March 2019
- [7] Ye-Kui Wang , Hendry , Maxim Sychev , Robert Skupin, Yago Sanchez, Karsten Suehring and Thomas Schierl, "AHG12: Signalling for tile and brick partitioning," JVET-N0857, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19-27 Mar. 2019
- [8] Gary Sullivan and Jens-Rainer Ohm, "Meeting Report of the 15th Meeting of the Joint Video Experts Team (JVET), Gothenburg, SE, 312 July 2019," JVET-O_Notes_dB, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3-12 July 2019
- [9] Lulin Chen, Ching-Yeh Chen, Yu-Wen Huang and Shaw-Min Lei "AHG17: [SYS-VVC] Signalling subpicture coded video sequence," JVET-N0073, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, Switzerland, 19-27 March 2019
- [10] Jianle Chen, Elena Alshina, Gary J. Sullivan, Jens-Rainer Ohm and Jill Boyce, "Algorithm Description of Joint Exploration Test Model 2," JVET-B1001, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 2nd Meeting: San Diego, USA, 20-26 February 2016
- [11] Gary Sullivan and Jens-Rainer Ohm , "Meeting Report of the 10th meeting of the Joint Video Experts Team (JVET), San Diego, US, 1020 Apr. 2018," JVET-J1000, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 10th Meeting: San Diego, US, 10-20 Apr. 2018
- [12] Xiang Li, Hsiao-Chiang Chuang, Jianle Chen, Marta Karczewicz, Li Zhang, Xin Zhao and Amir Said , "Multi-Type-Tree," JVET-D0117, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 4th Meeting: Chengdu, CN, 15-21 October 2016
- [13] Gary Sullivan and Jens-Rainer Ohm , "Meeting Report of the 11th meeting of the Joint Video Experts Team (JVET), Ljubljana, SI, 10-18 July 2018," JVET-K1000, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting: Ljubljana, SI, 10-18 July 2018
- [14] Chih-Wei Hsu, Tzu-Der Chuang, Ching-Yeh Chen, Yu-Wen Huang and Shaw-Min Lei, "CE1-related: Constraint for binary and ternary partitions," JVET-K0556, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting: Ljubljana, SI, 10-18 July 2018

필자 소개



남다운

- 2016년 ~ 현재 : 세종대학교 전자정보통신학과 학사과정
- 주관심분야 : 영상처리, 비디오 부호화 압축



한종기

- 1992년 : KAIST 전기및전자공학과 공학사
- 1994년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 3월 ~ 2001년 8월 : 삼성전자 DM연구소 책임연구원
- 2001년 9월 ~ 현재 : 세종대학교 전자정보통신공학과 교수
- 2008년 9월 ~ 2009년 8월 : University California San Diego(UCSD) Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-5036-7199>
- 주관심분야 : 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템