

움직임 제한 타일 기법을 활용한 사용자 시점 기반 360 영상 전송

손장우, 장동민, 정종범, 류은석
가천대학교

sjw6757@gc.gachon.ac.kr, dogzz9445@gc.gachon.ac.kr, uof4949@gc.gachon.ac.kr,
esryu@gachon.ac.kr

Viewport-Based 360 Degree Video Streaming using Motion-Constrained Tile Set

Jangwoo Son, Dongmin Jang, JongBeom Chung, Eun-Seok Ryu
Gachon University

요 약

가상 현실을 위한 360 영상 비디오 전송기술이 활발히 연구되고 있다. 그러나 현재 가상현실 기기의 컴퓨팅 연산능력과 대역폭은 고화질 360 영상을 재생하기에 한계가 있다. 이 한계를 극복하기 위해 본 논문은 High Efficiency Video Coding (HEVC)와 Scalability Extension of HEVC (SHVC)를 활용하여 타일 기반의 360 도 영상 전송 기법을 제안한다. 제안하는 HEVC 와 SHVC 인코더는 타일을 독립적으로 전송 할 수 있는 비트 스트림을 생성한다. 제안하는 추출기는 사용자 시점에 해당하는 타일의 비트 스트림을 추출한다. 제안하는 기법에 의해 추출된 SHVC 비트스트림의 기본계층은 전체화면을 나타내며, 강화계층은 사용자 시점에 해당하는 타일로 구성된다. 제안하는 HEVC 인코더를 사용할 때에는 저화질과 고화질을 따로 인코딩하여 고화질만 사용자 시점에 해당하는 타일을 추출한다. 전체화면을 고화질로 보내는 대신에 전체화면을 저화질로, 사용자화면을 고화질로 보내기 때문에 제안하는 기법은 디코더의 컴퓨팅 연산과 네트워크 bitrate 를 대폭 줄일 수 있다. 본 제안 기법의 실험 결과는 전체화면 전송 대비 47%이상의 bitrate 를 줄인다.

1. 서론

최근 많은 종류의 머리 장착형 기기와 360 카메라들이 시장에 출시되고 있으며, 가상현실 기술 발전에 따라 새로운 360 비디오 코딩과 파일 포맷 표준들이 활발히 연구되고 있다.

자연스러운 360 영상 재생과 사용자의 매스꺼움을 완화하기 위해 초 고화질 360 영상과 매우 낮은 지연시간이 요구된다 [1]. 하지만 현재 출시된 머리 장착형 기기의 스펙은 초 고화질 360 영상의 높은 컴퓨팅 연산을 완전하게 지원하지 못한다. 따라서 이 제한적인 조건을 극복하기 위한 360 영상의 효율적인 전송 방안이 필요하다. 실제로 국제 표준화 미팅에서 사용자 시점 기반 전송을 위해 Motion-constrained tile set (MCTS) 기법이 논의되었다 [2]. MCTS 기법은 현재 화면과 참조화면 간 동일한 위치에 있는 타일의 시간적 참조를 제한하여 타일의 독립적 전송을 보장한다. 본 논문은 SHVC 와 HEVC reference software (SHM 와 HM)에서 MCTS의 구현 기법을 제안한다. 본 논문의 구현은 Moving Picture Experts Group (MPEG) 표준에 채택되었으며 [3], 더 나아가 전체 타일에서 사용자 시점에 해당하는 타일을 추출하는 기법을 제안한다. 그림 1 은 서버와 클라이언트를 고려한 제안하는 시스템 개념도이다. 제안하는 SHM 과 HM 인코더는 독립적 타일 추출과 전송을 위해 시간적 움직임 정보를 제한하며, 추출을 위한 정보를 포함하고 있는 Extraction Information Sets (EIS) Supplemental Enhancement Information (SEI)

메시지를 생성한다 [4]. 제안하는 추출기는 클라이언트에게 사용자 시점 정보를 얻은 후 EIS SEI 메시지를 기반으로 원하는 타일에 해당하는 비트 스트림을 추출한다. 마침내 클라이언트는 추출된 비트 스트림을 디코딩한다.

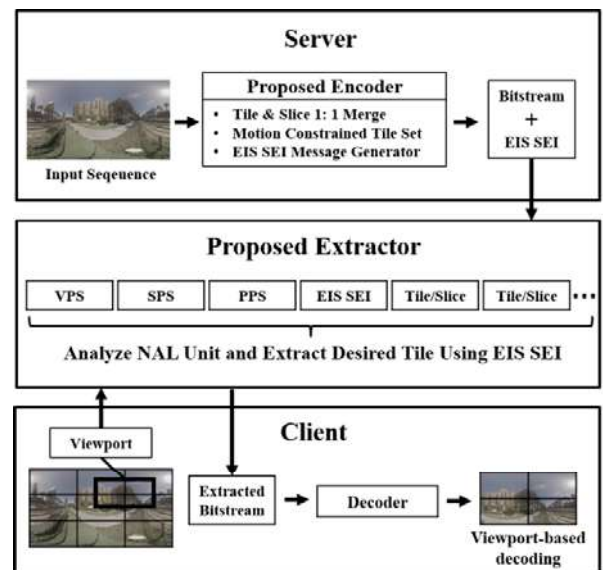


그림 1. 제안하는 시스템 서버/클라이언트 개념도

2. 관련 연구

360 화면에서 사용자가 바라보고 있는 영역은 전체 화면 중 일부이다. 따라서 사용자가 바라보고 있는 영역만 전송하면 대역폭은 상당히 줄어든다. 화면을 나누어 각각 인코딩하면 가장 간단하게 구현 할 수 있지만 번거로운 절차와 많은 저장 공간이 요구된다. 반대로 전체화면 하나의 비트 스트림에서 사용자 시점에 해당하는 비트 스트림만 전송하면 위의 단점을 극복 할 수 있지만 움직임 벡터가 고려되어야 한다. 왜냐하면 디코딩 시에 전송되지 않은 영역을 참조 할 수 있기 때문이다. Y. Sanchez 외 2 명은 참조 오류를 막기 위해 Generated Reference Picture (GRP)를 제안했다 [5]. GRP 의 모든 Prediction Unit (PU)는 선택적 타일을 전송할 때 발생하는 참조 문제의 보상 움직임 정보를 갖고 있다. 해당 연구는 GRP 를 통해 참조 오류를 수정했지만 여전히 GRP 를 생성해야 하는 오버헤드가 발생한다. MPEG 표준화 단체는 추가 화면 생성 없이 현재 화면의 움직임 벡터를 제한하는 MCTS 기법을 논의 했다 [2]. MCTS 기법은 인코더에서 시간적 움직임 정보를 제한하기 때문에 약간의 인코딩 효율 하락이 발생한다. 하지만 MCTS 기법이 적용된 비트 스트림은 타일의 독립성을 보장하여 추출이 용이하다. A. Zare 외 3 명은 HEVC 인코더를 MCTS 형태로 수정했다 [6]. 해당 연구는 3 개의 타일링 기법을 사용하였으며, 3%~6%의 인코딩 효율 손실 결과를 갖는다. 하지만 field of view (FOV)에 해당하는 타일을 전송할 경우 약 30%~40%의 bitrate 를 절약했다. 본 연구는 해당 연구와 다르게 SHVC 와 HEVC 인코더를 모두 수정하였으며 MCTS 의 구현 과정을 표준에 따라 상세하게 기술한다. 우리는 또한 EIS 메시지를 활용하여 MCTS 가 적용된 단일 비트 스트림을 추출하는 과정을 상세히 기술한다.

3. 움직임 제한 타일 기법을 활용한 사용자 시점 기반 360 영상 전송

3.1 디코딩 되지 않은 타일 영역의 시간적 참조

기존 SHM 인코더는 같은 계층 간에 화면 간 시간적 참조를 하며, 다른 계층 간에는 업-샘플링 필터를 이용하여 계층 간 참조를 한다 [7]. 이는 디코더가 전체화면을 디코딩 할 때 문제없이 진행되지만, 일부 타일만 디코딩 할 때 시간적 참조의 움직임 추정과 보상에서 문제가 발생한다. 그림 2 는 일부 타일만 디코딩 할 때 발생하는 시간적 참조 문제를 설명한다. 일부 타일만 디코딩 할 경우 디코딩 되지 않은 타일이 발생한다. 현재 화면 (PicEL t)이 참조 화면 (PicEL t-1)을 참조 할 때, 만약 인코더에서 생성된 움직임 벡터가 디코딩 되지 않은 타일을 참조 할 경우 디코딩 문제가 발생한다. 그래서 본 논문은 해당 문제를 해결하기 위해 인코더에서 시간적 움직임 정보를 수정한다. 즉, 시간적 참조의 움직임 벡터가 현재 화면과 참조 화면의 같은 위치 타일을 가리키면 현재 화면의 PU 는 보간법과 AMVP, MERGE 모드를 고려하여 참조화면의 PU 를 참조한다. 고려 사항들은 4 절에서 자세하게 설명한다. 이 경우 최적의 PU 를 찾기 위해 윌-웨이크 비용을 계산할 때 기존 인코더방식대로 시간적 참조와 계층 간 참조가 모두 고려된다. 반대로 현재 화면과 참조 화면의 다른 위치의 타일을 참조 하면 제안하는 인코더는 오직 계층간 참조만 수행한다. HEVC 의 경우에는 단일 레이어이기 때문에 계층 간 참조가 없다. 따라서 제안하는 HEVC 인코더는 다른 위치의 타일을 화면 간 시간적 참조 할 때 화면 내 공간적 참조를 수행한다.

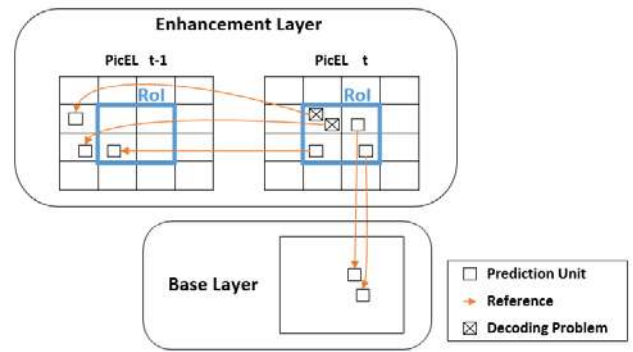


그림 2. SHVC 디코더에서 디코딩 되지 않은 타일을 참조 시 발생하는 문제

3.2 비트 스트림 추출을 위한 EIS SEI Message 생성

제안하는 인코더를 활용하여 MCTS 가 적용된 비트 스트림을 생성한 후 다음 단계는 사용자 시선에 해당하는 타일의 비트 스트림을 추출하는 것이다. 추출기는 Network Abstraction Layer (NAL) 단위 헤더를 분석하여 파라미터 셋, SEI 메시지, 슬라이스를 구별한다. 타일은 NAL 단위가 아니기 때문에 타일과 슬라이스를 1 대 1 로 합쳐 인코딩 해야 한다. 또한 인코더는 추출되는 타일에 따라 변화하는 대체 파라미터 셋을 전송해야 한다. 대체 되는 파라미터 셋 정보는 다음과 같다; VPS - 레벨과 티어, SPS - 화면 해상도, PPS - 타일 분할. 이를 대응하기 위해 MPEG 은 EIS SEI 메시지를 초안 하였다 [4]. 하나의 EIS SEI 메시지는 약 2,000 개의 EIS 를 포함 할 수 있기 때문에 거의 모든 추출 경우의 수를 포함 할 수 있다. 그림 3 은 하나의 EIS 가 포함하는 정보를 설명한다. 하나의 EIS 는 약 2,000 개의 MCTS set 과 하나의 파라미터 셋을 포함 한다. Slice Reordering 은 옵션이며, 올바르게 랜더링 하기 위해 타일과 합쳐진 슬라이스 주소를 변경하는데 사용된다 [8]. MCTS set 은 추출되는 타일 세트를 포함한다. 하나의 EIS 는 하나의 파라미터 셋을 포함하기 때문에 EIS 에 포함되는 MCTS Set 들은 동일한 해상도와 동일한 타일 분할 정보를 갖아야 한다. 예를 들어, 300x300 해상도의 화면이 100x100 의 9 개의 타일로 구성되어 있다고 가정해 보자. 하나의 EIS 는 1 개의 타일을 갖는 9 개의 MCTS set 을 포함 할 수 있다. 또한 또 다른 EIS 는 2x2 사각형 모양의 4 개 타일을 갖는 6 개의 MCTS set 을 포함 할 수 있다. 2x2 사각형 모양의 4 개 타일의 경우가 6 개인 이유는 Equirectangular projection (ERP)의 왼쪽과 오른쪽이 붙어 있기 때문에 4 개에서 2 개가 추가된다 [8]. 본 연구는 MPEG 의 EIS SEI 초안 문법을 실제 코딩으로 구현하였다.

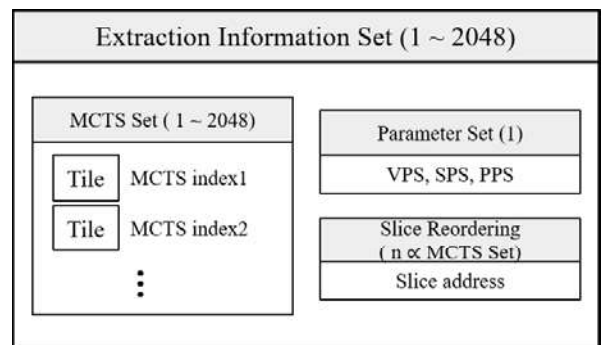


그림 3. EIS SEI 메시지 포함 정보

4. 구현

4.1 인코더: MCTS 를 위한 시간적 참조 수정

타일 기법은 병렬처리를 지원하기 위해 화면 내 공간적으로 독립적이다. 하지만 참조 화면은 이미 디코딩 되어 있기 때문에 타일 기법은 화면 간 시간적으로 독립적이지 않다. 따라서 시간적 참조 시 참조되는 PU 가 현재 화면과 참조 화면의 같은 위치 타일 안에 있는지 판단 할 때 움직임 추정과 보상에서 보간법이 고려되어야 한다. SHM 과 HM 은 시간적 참조 시 휘도 성분에 대하여 8 탭 필터를 사용한다 [9]. 8 탭 필터가 수평적으로 적용 될 때, 현재 픽셀 기준으로 왼쪽 3 개 픽셀과 오른쪽 4 개 픽셀이 보간 필터를 적용하기 위해 사용 된다. 수직적으로 적용 될 때에는 현재 픽셀 기준으로 위쪽 3 개 픽셀과 아래쪽 4 개 픽셀이 보간 필터를 적용하기 위해 사용 된다. 그림 4 의 (a)는 같은 위치 타일의 시간적 참조 시 보간법을 고려한 참조 수정 범위를 설명한다. 만약 그림 (a)의 빗금 친 영역을 시간적 참조하면 다른 타일의 픽셀이 보간 필터로 사용되기 때문에 디코딩 시 문제가 발생한다. 따라서 본 연구는 현재 화면과 참조 화면의 같은 위치 타일을 참조하더라도 빗금 친 영역을 참조 범위에서 제외하였다.

SHM 과 HM 은 시간적 참조 시 움직임 정보의 데이터를 줄이기 위해 Advanced Motion Vector Prediction (AMVP) 와 Merge 모드를 사용한다 [9]. 두개의 모드는 공간과 시간적 블록을 모두 후보로 사용한다. 여기서 현재 화면의 PU 기준으로 중앙과 오른쪽 아래의 블록이 시간적 후보로 사용된다. 만약 오른쪽 아래의 시간적 후보 블록이 현재 속해 있는 Coding Tree Unit (CTU)의 row 기준으로 넘어 간다면 해당 블록은 시간적 후보에서 제외된다 [9]. 하지만 현재 CTU 의 row 기준이 아닌 column 기준으로 넘어가면 시간적 후보로 고려되기 때문에 다른 위치 타일의 블록을 후보로 사용하여 디코딩 문제가 발생 할 수 있다. 그림 4 의 (b)는 타일 간 column 경계에서 다른 타일의 블록을 시간적 후보로 사용하는 예를 설명한다. (b)의 그림에서 H 블록은 다른 타일의 움직임 정보를 사용하기 때문에 타일의 독립적 전송을 보장 할 수 없다. 따라서 본 연구는 타일 간 column 경계에서 H의 시간적 후보를 제외시켰다.

본 연구가 제안하는 소스는 HM 16.18 에서 #if MCTS_ENC_CHECK 지시문을 통해 확인 할 수 있다 [10].

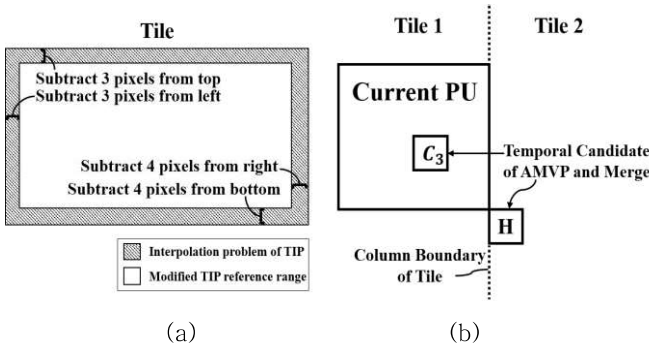


그림 4. (a) 보간법을 고려한 시간적 참조 범위 수정, (b) AMVP 와 Merge 모드에서 시간적 후보 제한

4.2 추출기: 사용자 시점 기반 타일 추출

추출기는 MCTS 가 적용된 인코더로부터 생성된 비트 스트림에서 사용자 시점에 해당하는 타일의 비트 스트림을

추출하는 역할을 한다. 그림 5 는 추출기의 기능 흐름도를 보여준다. 입력 값은 target EIS id, MCTS set id, highest temporal id 이다. 입력 값 설정 후 추출기는 NAL 단위로 분석을 시작한다. 기존 PPS 를 분석할 때, 추출기는 화면단위로 NAL 유닛을 분석하기 위해 기존 화면을 구성하는 타일 개수를 구한다. 다음으로 추출기는 인코더에서 생성한 EIS SEI 메시지를 분석하여 기존 파라미터 셋을 target EIS id 값에 맞는 대체 파라미터 셋으로 바꾼다. 이후 추출기는 원하는 타일을 추출하기 위해 타일/슬라이스의 NAL 헤더를 분석하여 $nuh_temporal_id_plus1 - 1$ 의 값이 target highest temporal id 값보다 작은 값을 선별한다. temporal id 는 대역폭에 적용적인 대응을 할 수 있게 한다. NAL 유닛이 SHM 인코더에서 생성된 경우에는 강화 계층인지 구별하기 위해 nuh_layer_id 가 0 보다 큰지 판단한다. temporal id 와 nuh_layer_id 조건을 만족하면 추출기는 target MCTS set id 에 해당하는 타일/슬라이스를 추출하고, 추출한 타일/슬라이스 헤더의 $first_slice_segment_in_pic_flag$ 와 $slice_segment_address$ 를 수정한다. 만약 타일/슬라이스가 화면의 첫번째 타일/슬라이스일 경우 $first_slice_segment_in_pic_flag$ 값을 1 로 설정하고 그렇지 않으면 0 으로 설정한다. 만약 EIS SEI 메시지의 $slice_reordering_flag$ 값이 1 인 경우, $slice_segment_address$ 값을 EIS SEI 메시지의 $output_slice_segment_address$ 로 설정한다. 그 외의 경우에는 $slice_segment_address$ 값을 래스터 주사 방식으로 설정한다. 추출기는 위의 방식으로 화면 단위로 모든 NAL 유닛을 분석 후 추출된 비트 스트림을 생성하고 종료한다.

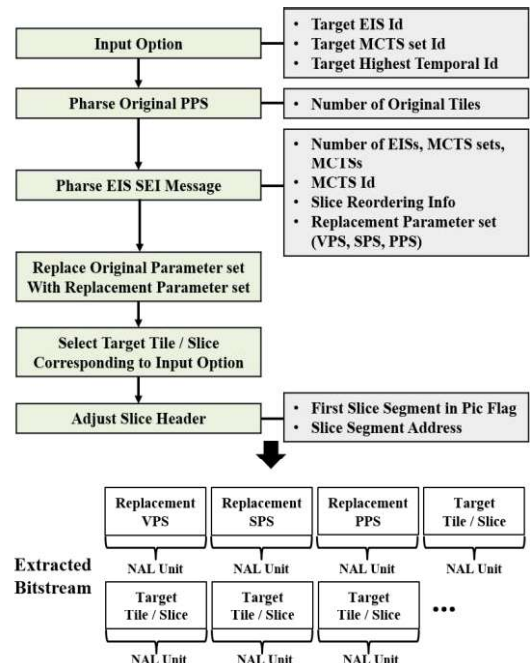


그림 5. 추출기의 기능 흐름도

5. 실험 결과

본 연구는 Joint Video Exploration Team (JVET)에서 선정된 테스트 시퀀스를 사용하였으며 해당 시퀀스들은 Random Access (RA) 코딩으로 인코딩 되었다 [11]. 본 연구는 Field of View (FOV)가 90 ~ 110 도 입을 감안하여 타일을 균등하게 3x3 9 등분으로 분할하였다. 따라서 최소 1 개부터 최대 4 개의 타일 까지 사용자 시점을 포함할 수 있다.

본 연구는 9 개의 경우의 수로 1 개 타일을 추출하고, 6 개의 경우의 수로 사각형 모양의 4 개 타일을 추출하였다.

제안하는 SHM 과 HM 의 bitrate 는 각각 평균 8%와 11% 증가 하였으며, PSNR 은 각각 평균 0.04dB 와 0.05dB 감소하였다. 타일의 독립적 전송을 보장하기 위해 시간적 움직임 정보를 제한했기 때문에 bitrate 와 PSNR 에서 약간의 손실이 발생하였다.

표 1 은 전체 화면을 보낸 bitrate 와 비교하여 제안하는 SHM 인코더를 사용하였을 때 4 개와 1 개의 타일을 보낸 경우 절약되는 bitrate 결과 값을 보여준다. 표 2 의 경우에는 표 1 과 측정 방식이 같은 대신 제안하는 HM 인코더를 사용하였을 때 절약되는 bitrate 결과 값을 보여준다. 제안하는 SHM 인코더를 사용하여 4 개 타일을 전송 할 경우 평균 48%의 bitrate 를 줄였으며, 1 개 타일을 전송 할 경우 최소 75%부터 최대 97%의 bitrate 를 줄였다. 제안하는 HM 인코더를 사용하여 4 개의 타일을 전송 할 경우 평균 47%의 bitrate 를 줄였다.

Name	4 tiles bit-rate saving (average)	1 tile bit-rate saving (min, max)
<i>KiteFlite</i>	49%	77%, 96%
<i>Harbor</i>	46%	63%, 98%
<i>Trolley</i>	50%	80%, 98%
<i>GasLamp</i>	48%	80%, 97%
<i>Average bit-rate saving</i>	48%	75%, 97%

표 1. 제안하는 SHM 인코더를 사용하여 전체화면 중 4 개와 1 개 타일을 전송할 때의 bitrate 절약률

Name	4 tiles bit-rate saving (average)	1 tile bit-rate saving (min, max)
<i>KiteFlite</i>	49%	78%, 96%
<i>Harbor</i>	46%	63%, 98%
<i>Trolley</i>	49%	80%, 98%
<i>GasLamp</i>	47%	79%, 97%
<i>Average bit-rate saving</i>	47%	75%, 97%

표 2. 제안하는 HM 인코더를 사용하여 전체화면 중 4 개와 1 개 타일을 전송할 때의 bitrate 절약률

구현된 플레이어는 FFmpeg 의 OpenHEVC 디코더를 활용하여 실시간 디코딩을 지원한다 [12]. 그림 6 은 IEEE 802.11ad (60GHz mmWave) 통신을 활용하여 모바일에서 9 개의 타일 중 하나의 타일을 추출한 비트 스트림을 디코딩하고 렌더링하는 구현 장면이다. 데모 비디오 링크는 다음과 같다; <https://youtu.be/--zDLYEau54>.



그림 6. mmWave 통신을 이용한 모바일 VR 플레이어 구현 (3840x1920 의 9 타일 중 1280x640 추출된 1 타일)

5. 결론

본 연구는 사용자 시점 기반 SHVC 와 HEVC 타일 기법 전송을 제안한다. SHVC 를 활용하여 기본계층은 전체화면을 인코딩 및 디코딩 하며 강화계층은 시간적 참조 시 제한하여 인코딩 후 사용자 시점에 해당하는 타일을 디코딩 한다. HEVC 인코더는 다른 위치 타일 시간적 참조 시 화면 내 인코딩을 수행한다. 제안하는 기법은 시간적 참조 시 AMVC 와 Merge 의 시간적 후보와 참조 범위를 제한하기 때문에 인코딩 효율에서 약간의 손실이 발생하지만 타일을 독립적으로 전송 할 수 있다. 제안하는 SHM 과 HM 인코더를 활용하여 전체 9 개의 타일 중 4 개의 타일을 전송할 때 각각 평균 48%와 47%의 bitrate 를 줄였으며 1 개의 타일을 전송 할 경우 최소 75%부터 최대 97%의 bitrate 를 줄였다.

참고문헌

- [1] M. Champel, T. Stockhammer, T. Fautier, E. Thomas and R. Koenen. 2016. Quality Requirements for VR. Technical Report ISO/IEC JTC1/SC29/WG11/ MPEG 116/m39532.
- [2] Y. K. Wang, Hendry and M. Karczewicz. 2016. Viewport dependent processing in VR: partial video decoding. Technical Report ISO/IEC JTC1/SC29/WG11/ MPEG 116/m38559.
- [3] R. Skupin, Y. Sanchez, K. Suhring, T. Schierl, E. S. Ryu and J. Son. 2017. Temporal MCTS Coding Constraints Implementation. Technical Report ISO/IEC JTC1/SC29/WG11/MPEG 120/ m41626.
- [4] J. Boyce, A. Ramasubramanian, R. Skupin, Gary J. Sullivan, A. Tourapis and Y. K. Wang. 2017. HEVC Additional Supplemental Enhancement Information (Draft 4). Technical Report ISO/IEC JTC1/SC29/WG11/JCTVC-AC1005.
- [5] Y. Sánchez, R. Skupin and T. Schierl. 2017. Video processing for panoramic streaming using HEVC and its scalable extensions. Multimedia Tools and Applications. 5631- 5659.
- [6] A. Zare, A. Aminlou, M. Hannuksela and M. Gabbouj. 2016. HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications. In Proceedings of the 24th ACM Multimedia Conference. Netherlands, 601-605.
- [7] J. Boyce, Y. Ye, J. Chen and A. K. Ramasubramanian. 2015. Overview of SHVC: scalable extensions of the high efficiency video coding standard. IEEE Transactions on Circuits and Systems for Video Technology. 26, 1(2015), 20-34.
- [8] R. Skupin and Y. Sanchez. 2017. MCTS extraction with optional slice reordering. Technical Report ISO/IEC JTC1/SC29/WG11/MPEG 119/m40902.
- [9] B. Bross, P. Helle, H. Lakshman and K. Ugur. 2014. Inter-Picture Prediction in HEVC. Integrated Circuits and Systems. Springer, 113-140.
- [10] High Efficiency Video Coding (HEVC) reference software HM. 2017. Retrieved from <https://hevc.hhi.fraunhofer.de/>
- [11] J. Boyce, E. Alshina, A. Abbas and Y. Ye. 2017. JVET common test conditions and evaluation procedures for 360° video. Technical Report ISO/IEC JTC1/SC29/WG11/JVET-H1030.
- [12] Github. 2018. FFmpeg. Retrieved from <https://github.com/FFmpeg/FFmpeg>.