

## 사용자 시점 기반 360 영상을 위한 렌더러 구현

장동민, 손장우, 정종범, 류은석  
가천대학교

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

### Implementing Renderer for Viewport Dependent 360 Video

Dongmin Jang, Jang-Woo Son, JongBeom Jeong, Eun-Seok Ryu  
Gachon University

#### 요 약

본 논문에서는 실시간 고화질 360 영상 전송을 위해 사용자 시점에 기반한 타일 분할 기법을 적용하고 360 영상을 가상현실 기기 화면에 표현하여 주관적 화질 평가를 위한 플랫폼을 구현한다. 사용자 시점에 기반한 고화질 360 영상 전송을 위한 방안으로, 분할된 영상에서 전송되지 않은 영상으로의 움직임 참조 문제를 해결하기 위해 적용된 MCTS (motion constrained tile sets) 기술과 실시간으로 사용자 시점에 위치한 타일들만 추출할 수 있도록 미리 구성된 타일 정보들을 포함하는 EIS (extraction information sets) SEI, 사용자 시점에 위치한 타일 정보만 추출해내고 영상을 분할 및 추출해주는 Extractor, 실제 추출된 영상 정보를 이용해 가상현실 기기 화면에 표현하는 방법에 대한 구현 내용을 설명한다. 따라서 제안된 구현물을 기반으로 고화질 360 영상 전송을 수행하면, 사용자 시점 영역의 영상만 전송하여 불필요한 영상 전송을 하지 않게 되어 화질 대비 낮은 대역폭의 향상된 실시간 전송 영상을 얻을 수 있다.

#### 1. 서론

최근 몰입형 실감 미디어인 360 영상의 이용이 급속도로 증가하고 있다. 이러한 360 영상을 높은 몰입감으로 보기위해 최소한 UHD 급의 고화질 영상이 요구된다 [1]. UHD 동영상의 전송은 높은 대역폭을 요구하게 되고, 따라서 고해상도 360 영상의 효율적인 전송 기법이 요구되고 있는 실정이다.

Motion Constrained Tile Sets (MCTS) 는 영상의 각 타일을 공간적 및 시간적 독립성을 적용해 부호화하는 기법으로 각 타일을 독립적으로 전송할 수 있다 [2]. 각 타일을 독립적으로 전송함으로써 고화질 360 영상에서 사용자 시점에 해당하는 타일만 추출하여 전송하고, 사용자는 해당하는 영역만 복호화 및 구체 매핑, 화면에 렌더링하여 기존 영상 대비 낮은 대역폭을 가지고 복호화 속도를 향상 시킬 수 있다.

일반적으로 360 동영상 플레이어는 MCTS 기법이 적용되지 않고 표준화된 H.264/AVC 와 H.265/HEVC 두 가지 동영상 압축기술을 주로 사용하여 고화질 360 영상 압축 처리를 하고 있다 [3]. 따라서, 현재 실시간으로 타일 기반 압축된 영상을 복호화하여 모니터나 머리 착용 디스플레이에 렌더링할 수 있는 플랫폼이 없다. 압축된 영상을 렌더링을 하기 위해서는 비트스트림을 복호화하는 과정에서 생기는 타일 정보와 해상도, 프레임에 관련된 부가적인 메타데이터를 플랫폼에서 처리할 수 있는 데이터로 구성이 필요하다. 렌더러는 영상과 함께 전송된 메타데이터를 이용해 특정 타일만 구체에 매핑하고 사용자 화면에 렌더링 가능해진다.

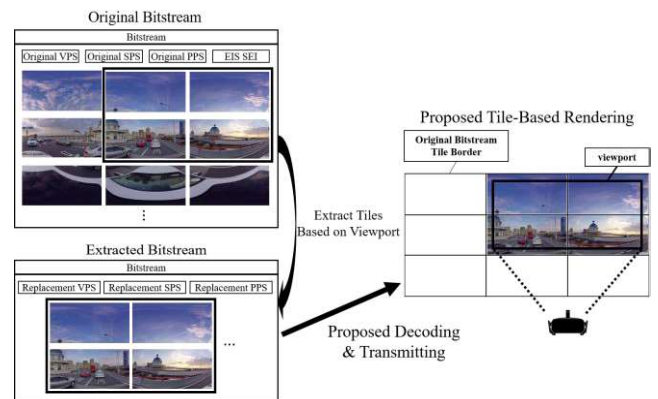


그림 1. 제안하는 전송 및 렌더링 구조

본 논문에서는 타일 기반으로 추출된 영상을 렌더링하기 위하여 MCTS 와 EIS SEI 정보를 통해 부가적인 메타데이터를 구성하고 구체에 각 타일을 매핑하는 방법을 제안한다. 그리고나서, 렌더러의 설계 및 구현에 대해 설명한다. 그림 1 은 MCTS 가 적용된 영상에서 부가적인 정보를 포함한 전송과 해당 타일만 추출한 영상을 복호화하는 과정, 타일 기반 렌더링 방식에 대한 제안을 보여준다.

본 논문의 구성은 다음과 같다. 2 절에서는 MCTS, EIS SEI, Extractor 에 대한 설명하고 3 절에서는 렌더러를 위한 타일 기반 전송 이슈에 대한 내용을 토의한다. 4 절에서는 렌더러 설계 및 구현 결과물에 대한 설명을 진행한다. 마지막으로 5 절에서는 본 논문에 대한 결론을 맺는다.

## 2. 관련 연구: 타일 기반 영상 전송

### 2.1 Tile Encoding Based on MCTS

기존의 HEVC 와 SHVC 에 정의된 타일은 병렬 처리를 지원하기 위해 공간적 예측 부호화를 각 타일별로 독립적 예측을 한다. 하지만, 각 타일들은 시간적 예측 부호화에 대한 독립성을 가지고 있지 않아, 이전 화면의 모든 타일들이 복호화 되어 있어야 현재 화면의 타일들을 복호화 할 수 있는 문제점이 있다. MCTS 는 일시적으로 인코더에서 시간적 움직임 추정을 공간적 움직임 추정으로 제한하여 시간적 및 공간적으로 독립적인 타일로 구성된 비트스트림 파일을 생성한 후, 타일과 타일을 구별하기 위해 NAL 단위의 슬라이스를 타일과 1:1 로 병합한다 [2].

#### 2.1.1 움직임 추정 화소 간의 예측오차 거리 수정

SHM 및 HM 에서 현재 타일이 동일한 위치의 타일을 시간적으로 참조 할 때, Eight-Tap 필터를 사용하여 휘도 성분에 대한 보간을 한다. Eight-Tap 필터를 수평으로 적용될 때, 현재 픽셀의 좌측으로 3 픽셀 및 우측으로 4 픽셀이 사용된다. 수직으로 적용될 때, 현재 픽셀의 위 측의 3 픽셀과 아래 측의 4 픽셀이 사용된다. 따라서, 시간적으로 동일한 위치의 타일에 위치한 PU 를 참조할 때에도, 픽셀 간 보간이 되는 영역이 다른 타일을 참조하지 않도록 고려하여야 한다. 그림 2 는 Temporal Inter Prediction (TIP) 에서 동일한 위치의 타일을 참조할 때, 문제가 나타나는 보간 영역을 보여준다. PU 가 보간 문제가 일어나는 영역을 참조하면 PU 가 다른 타일의 픽셀을 사용하여 보간하기 때문에, 타일을 독립적으로 전송할 수 없다. 따라서, 문제가 나타나는 보간 영역은 TIP 가 참조할 수 있는 범위에서 제외되어야 한다.

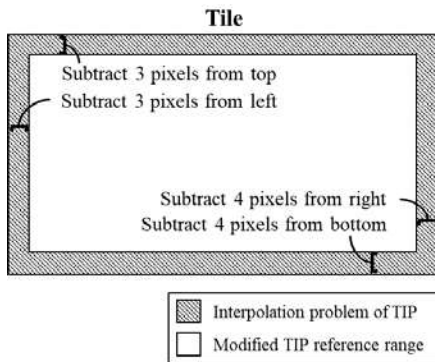


그림 2. TIP 에서 동일한 위치의 타일을 참조할 때, 문제가 나타나는 보간 영역

#### 2.1.2 타일 경계에서 AMVP 와 MERGE 후보 결정

SHM 및 HM 은 개선된 움직임 예측 방식인 AMVP 및 MERGE 방법을 사용하여 화면 간 예측에서 움직임 오차를 줄인다. 두 가지 방법 모두 공간적 및 시간적 후보를 찾아서 사용한다. 앞서 언급한대로 참조하는 후보 블록은 동일한 타일 내의 후보를 사용하도록 고려해야 한다. 기본적으로, 현재 PU 의 우측 하단에 있는 블록이 현재 Coding Tree Unit (CTU) 의 수평 경계를 넘어서면 후보로 사용되지 않는다. 그러나 참조하는 후보 블록이 수직 경계 밖으로 나갈 경우에는 문제가 생긴다. 그림 3 과 같이 H 후보 블록이 타일의 수직 경계 밖으로 나갈 때, AMVP 및 MERGE 방식에 의해 H 후보 블록이 선택되면 문제가 생길 수 있다.

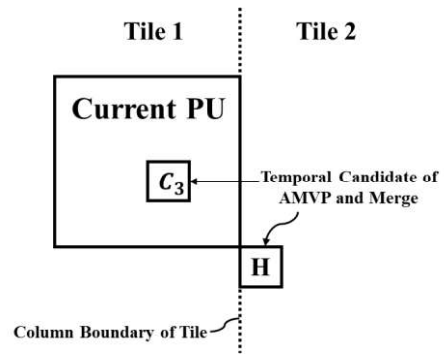


그림 3. 타일 간 수직 경계에서 후보 결정 문제

### 2.2 EIS SEI Message

Extraction Information Sets (EIS) SEI Message 는 MCTS 기술이 적용된 압축 영상으로부터 사용자가 원하는 Tile 을 추출해낼 수 있도록 임의로 구성된 파라미터 셋을 제공한다 [4]. 그림 4 는 EIS SEI 메시지 생성을 위해 위의 세 칸은 각각 SEI 메시지 정의와 생성, 비트스트림을 쓰는 기능을 한다.

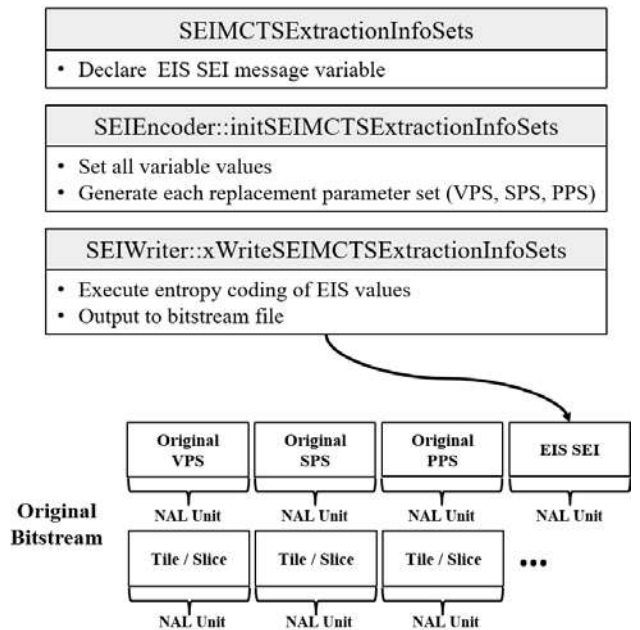


그림 4. EIS SEI 메시지 생성을 위한 주요 클래스 및 기능

### 2.3 Tile Extractor

Tile Extractor 는 원본 비트스트림과 EIS SEI 메시지를 파싱하고 VPS, SPS, PPS 일부 정보를 대체한다. 그림 5 는 Extractor 의 기능을 나타낸다. 세부 기능은 원본 비트스트림에서 Tile 개수, EIS 및 MCTS Sets, 대체 파라미터 셋을 파싱하고 VPS, SPS, PPS 의 정보를 대체한다. SPS 에서는 휘도 성분의 높이와 너비에 대한 정보를 타일의 높이와 너비로 대체하고 PPS 에서는 타일 정보를 대체한다.

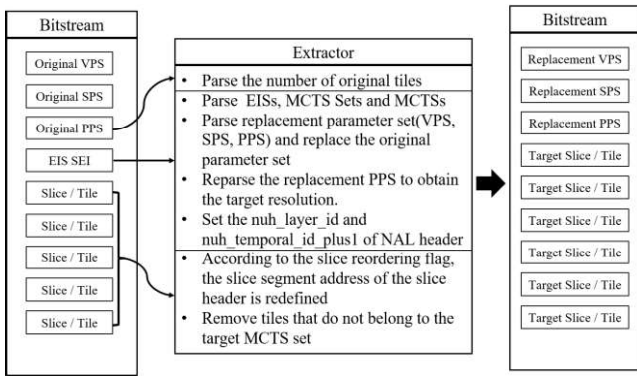


그림 5. Extractor 의 기능

### 3. 렌더러를 위한 타일 분할 및 전송 구조

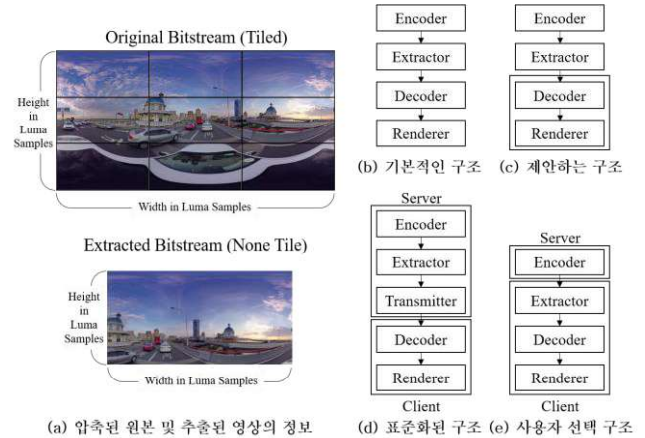
본 논문에서는 가상현실기인 Oculus Rift 를 이용하여 실험을 진행하였는데 Oculus Rift 는 수평 시야각과 수직 시야각 모두 100° 이다 [7]. 또한, 논문에서 이용한 영상은 Equirectangular Projection (ERP) 360 영상을 이용했으며 영상을 9 개의 타일로 분할하여 실험을 진행하였다.

#### 3.1 Extractor 를 이용한 전송 기법

렌더러 설계 및 구현에 앞서 진행한 사항으로, 그림 6 의 (a)와 같이 원본 비트스트림이 Extractor 에 의해 대체된 VPS, SPS, PPS 정보로는 대체된 비트스트림이 원본 비트스트림의 몇 번째 타일인지 유추하기 어렵다. 동기화되지 않은 플랫폼 상에서 사용자 시점 정보를 Extractor 로 전달하고 Extractor 가 대체된 비트스트림을 보냈을 때, 입력 버퍼 상에 위치한 NAL 단위의 비트스트림이 현재 사용자 시점이 해당하는 타일과 동일한 위치의 타일이 아닐 수 있다. 그림 6 의 (d)와 같이 미디어 전송 표준인 MPEG-DASH, MPEG Media Transport (MMT)로 구현된 서버-클라이언트 구조에서는 모든 비트스트림이 구성이 되어있어 사용자 시점에 해당하는 비트스트림을 동기화되어 전송된다. 따라서, 타일 기반 360 동영상 전송을 시험하려면 미디어 전송 표준에 의존되어 구현되어야 한다는 어려움이 있다.

그림 6 의 (c)는 제안하는 구조로 사용자 시점 처리 정보를 받아 추출해주는 그림 6 의 (b)와 같은 기본적인 구조에서 원본 비트스트림의 해상도 정보, 원본 비트스트림의 총 타일 개수, 대체된 비트스트림의 타일 개수, 대체된 비트스트림이 원본 비트스트림의 몇 번째 타일들로 구성되었는지에 대한 추가적인 정보들을 전송한다. 그리고 디코더와 렌더러의 정보를 유동적으로 사용할 수 있게 구성해 영상과 추가적인 정보들을 처리한다.

그림 6 의 (e)는 사용자 쪽에서 Extractor 를 이용하여 타일 분할을 할 수 있는 구조로써 네트워크 대역폭을 좀 더 사용하는 불이익이 있지만, 디코딩이나 렌더링 시에 복잡도를 줄일 수 있다. 향후 더 나아가, 사용자 시점에 대한 네트워크 지연이나 CPU 나 GPU 컴퓨팅 파워, 5G 기술 기반한 Mobile VR 을 고려해 구조를 설계하고 구현할 수 있다.



(a) 압축된 원본 및 추출된 영상의 정보 (b) 기본적인 구조 (c) 제안하는 구조 (d) 표준화된 구조 (e) 사용자 선택 구조

그림 6. 추출된 영상과 원본 영상이 가진 정보의 차이와 Extractor 를 포함한 시스템 구조도

#### 3.2 렌더러를 위한 타일 분할

분할된 타일들은 각각 시간적 및 공간적 독립성을 가지지만, 하나의 HEVC 상에서 디코딩하기 위해서는 분할된 타일들은 직사각형의 형태여야 한다. 따라서, 하나의 HEVC 를 이용해 단일 프로세스로 디코딩과 렌더링을 진행하는 경우에는 시점에 위치하지 않은 타일을 포함할 수 있다.

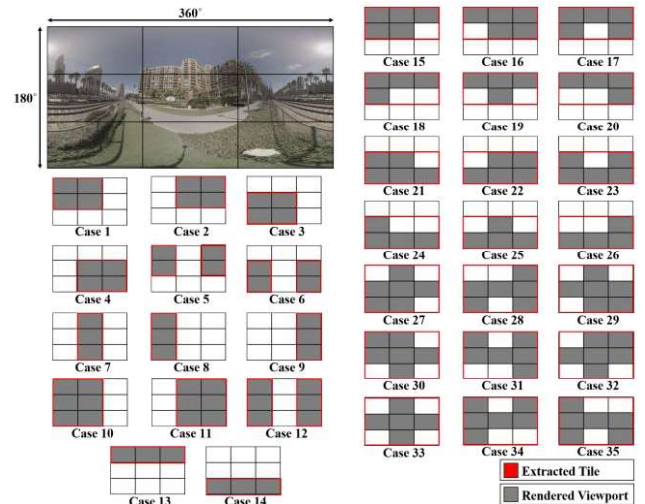


그림 7. HMD 상에 렌더링 되는 Tile 들과 Extracted Tile 들

그림 7 을 보면, 시점을 고려해 HMD 상에 렌더링 되는 Tile 들의 구성은 35 가지가 있다. 시점에 위치한 타일들을 기반으로 Extract 를 할 경우에는 직사각형의 형태 전송을 고려해야한다. 위의 붉은색 영역이 이런 점을 고려해 Extract 되어야 하는 타일들을 나타낸 영역이다. 따라서, 분할과 렌더링 시 그림 7 의 붉은 영역으로 구성되는 17 가지의 경우 (그림 7. Case1~14, Case15, Case21, Case27)를 고려한다.

### 4. 구현 내용 및 결과

본 연구에서는 HM 16.16 과 Oculus Rift SDK, OpenGL 을 사용하여 구현을 진행하였으며, MCTS 구현과 EIS SEI 구현, Extractor 구현은 [8] 의 구현물을 사용하였다.

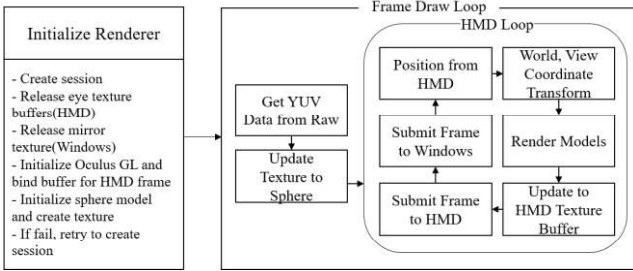


그림 8. 렌더러 기본적인 기능 구조도

본 논문에서 렌더러는 셰이더를 사용하지 않고 OpenGL 의 고급기능을 포함한 GLU 라이브러리를 사용해 구체를 이용하며, ERP 360 영상을 구체에 매핑하는 방식을 사용한다 [6, 7, 9]. 구체는 OpenGL 의 모델인 gluSphere 를 이용해 slice, stack 을 각각 40 분할한다. 정의되고 생성된 텍스처에 디코딩된 ERP 360 영상 한 프레임의 YUV 420 픽셀 값을 RGB 픽셀 값으로 변환하여 텍스처를 업데이트한다.

그림 8 과 같이 세션 생성, HMD 양안에 쓰이는 버퍼들, 모니터 디스플레이에 나타날 미리 버퍼들, Oculus SDK 에서 재정의된 OpenGL Sphere 모델의 생성과 텍스처 버퍼 생성과 텍스처 바인딩을 한다. 그 후, 프레임을 그리는 함수 루틴을 실행한다. 프레임을 그리는 루틴은 프레임을 불러와 업데이트 하는 부분과 Oculus SDK 지원하는 HMD 프레임을 그리는 루틴 부분으로 나눌 수 있다. Oculus Rift 는 약 90Hz 정도로 디스플레이가 주사하고 동영상 프레임은 그보다 낮은 속도로 재생되기 때문에, HMD 프레임 그리는 루틴과 따로 일정 주기마다 영상을 읽어 들이는 함수를 구현한다.

본 논문에서 제안하는 타일 기반 텍스처 매핑은 원본 비트스트림의 높이와 너비 값을 이용해 텍스처를 생성한다. 그리고 각 프레임마다 9 개의 타일로부터 분할 및 구성된 타일 높이와 너비 값과 타일 번호를 이용한다. 그림 10 의 오른쪽 그림과 같이 현재 뷰포트 영역의 타일만 텍스처 버퍼 영역에 그리고 만들어진 텍스처를 glSphere 객체에 업데이트한다. 그림 11 은 모든 과정을 진행하여 HMD 에서 렌더링한 화면을 나타낸다.

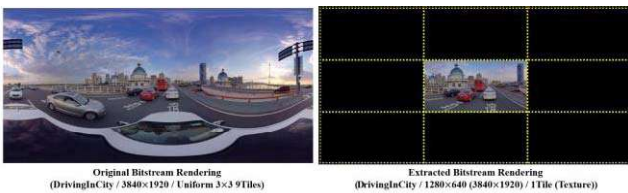


그림 10. 원본 영상과 타일 1 개 영상



그림 11. 렌더링 화면 (Trolley\_8192x4096)

## 5. 결론

본 연구에서는 사용자 시점 기반 타일 분할 기법과 렌더러 구현에 대해 설명한다. 구현물은 Oculus Rift 와 OpenGL 을 이용하여 구현하였으며, 타일로 분할된 영상을 구체에 렌더링하였다. 본 논문에서 인용한 MCTS 기술은 현재 표준화되고 있는 MPEG-I 3DoF와 6DoF 기술에도 적용이 가능하다. 따라서, 사용자 시점 기반 타일 분할 기술과 참조 소프트웨어를 이용한 렌더러를 연구 개발 및 구현해 나갈 예정이다.

## 참고문헌

- [1] Mary-Luc Champel, Thomas Stockhammer, Thierry Fautier, E Thomas, R. Koenen. 2016. Quality Requirements for VR. 116th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG 116/m39532.
- [2] Robert Skupin, Yago Sanchez, Karsten Suehring, Thomas Schierl, Eun-Seok Ryu, Jangwoo Son. 2017. Temporal MCTS Coding Constraints Implementation. 120th MPEG meeting of ISO/IEC JTC1/SC29/WG11, MPEG 120/m41626.
- [3] "Facebook 360 Video", Retrieved 18 May 2018, from <https://facebook360.fb.com/>.
- [4] Jill Boyce, Adarsh Ramasubramanian, Robert Skupin, Gary J. Sullivan, Alexis Tourapis, Ye-Kui Wang. 2017. HEVC Additional Supplemental Enhancement Information (Draft 4). 29th JCT-VC meeting of ISO/IEC JTC1/SC29/ WG11, JCTVC-AC1005.
- [5] Ye-Kui Wang, Hendry, Marta Karczewicz. 2016. Viewport dependent processing in VR: partial video decoding. 116th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG 116/ m38559.
- [6] Bassbouss, Louay, Stephan Steglich, Martin Lasak. "High quality 360° video rendering and streaming." NEM Summit proceedings. 2016.
- [7] "Oculus rift", Retrieved 18 May 2018, from <https://www.oculus.com/rift>.
- [8] Jang-Woo Son, Dongmin Jang, Eun-Seok Ryu. (2018). Implementing Motion-Constrained Tile and Viewport Extraction for VR Streaming. ACM Network and Operating System Support for Digital Audio and Video 2018 (NOSSDAV2018).