

## 각도정보 기반 360 VR 콘텐츠 내 사용자 시점예측기법 및 비트율 할당 방법

정은영 서봉석 현창종 김동호

서울과학기술대학교

{jeunyoung, sbs91, dksshddl, dongho.kim}@seoultech.ac.kr

An Method of Viewport Prediction and Bitrate Allocation  
based on Angle Information in 360 VR Contents

Jeong, Eunyoung Seo, Bong-seok Hyun, Chanjong Kim, Dong Ho

Seoul National University of Science and Technology

## 요약

360 VR 콘텐츠는 기존의 선형적인 일반 영상에 비해 사용자에게 더 많은 정보와 높은 몰입감을 제공할 수 있어 의학, 군사, 교육, 게임 등 다양한 분야에서 활용되고 있다. 최근에는 모바일 기기의 성능 향상과 통신기술의 발달에 힘입어 모바일 네트워크를 사용한 360 VR 콘텐츠 소비가 증가하는 추세이다. 모바일 네트워크는 대역폭이 한정적이고 가변적인 특성이 있어 이를 통해 용량이 큰 360 VR 콘텐츠 전송 시 초기 접속 지연 및 재생 끊김이 발생하여 사용자의 만족도를 감소시킬 수 있다.

이에 본 논문은 위에 언급한 문제를 해결하기 위해 360 VR 콘텐츠 전송 시 전체 요구대역폭을 감소시키고 사용자 초기 접속 속도를 향상시키는 것을 목표로, 360 VR 콘텐츠의 지오메트리 값과 사용자의 요(i.e. yaw)값을 활용하여 각도 기반으로 사용자의 현재 시점에 해당하는 타일을 확인하고 해당 타일에 높은 비트율을 할당하는 방법 및 웹 기반 전송에 대해 연구·개발하였다. 이를 위하여 웹 기반 3D 렌더링 API 인 WebVR API, HTTP Adaptive Streaming 기술의 표준 MPEG-DASH의 dash.js API를 활용하여 개발하고, 성능 확인 실험을 통해 요구대역폭 감소, 클라이언트 접속 속도 향상을 제시한다.

## 1. 서론

360 VR(Virtual Reality) 콘텐츠는 기존의 선형적인 일반 영상에 비해 사용자에게 더 많은 정보와 높은 몰입감을 제공하며 의학, 군사, 교육, 게임 등 다양한 분야에서 활용되고 있다[1]. 최근에는 모바일 기기의 성능 향상과 통신기술의 발달에 힘입어 모바일 네트워크를 통한 360 VR 콘텐츠 소비가 증가하는 추세이다. Sketchfab의 360 VR 콘텐츠 소비 행태 조사에 따르면 76.32%의 응답자가 모바일 기기로, 23.68%의 응답자가 데스크톱으로 시청하는 것으로 나타났다[2]. 이러한 추세로 미루어 보았을 때 모바일 네트워크에서 360 VR 콘텐츠를 감상하는 수요는 계속해서 증가할 것으로 예상된다.

360 VR 콘텐츠 내 사용자의 실 시청 영역(Viewport)은 일부분에 불과하다. 그러나 전송 시 360도로 구성된 공간 전체로 전송되므로 시청되지 않는 영역을 전송하는 만큼의 불필요한 요구대역폭이 발생한다. 이는 대역폭이 한정적이고 가변적인 모바일 네트워크에서 긴 사용자 접속 시간, 재생 중단과 같은 문제를 야기해 사용자의 만족도를 감소시키는 결과를 초래할 수 있다. 이는 모바일 네트워크를 통한 360 VR 콘텐츠 소비가 증가하는 추세이므로 사용자의 만족도 향상을 위해 해결해야 할 과제 중 하나이다. 이러한 문제를 해결하기 위해 사용자의 시점을 예측하여 시점에 해당하는 영역에 높은 비트율(bitrate)을 할당하고, 그 외 영역은 낮은 비트율을 할당하는 시점 예측 기법 관련 연구가 진행되고 있다[3].

본 논문은 사용자의 오리엔테이션 중 요(Yaw) 값과 영상의 지오메

트리(geometry) 값을 활용해 사용자의 현 시점에 해당하는 영역 정보를 획득하고, dash.js API를 활용해 순차적으로 비트율을 할당하는 알고리즘을 제안한다. 논문의 구성은 다음과 같다. 배경연구로 본 논문의 전송 방법인 HAS(HTTP Adaptive Streaming) 기술의 표준 MPEG-DASH(Dynamic Adaptive Streaming over HTTP)와 전체 요구대역폭 감소를 위한 연구 주제인 시점 예측 기법에 대해 기술한다. 그 다음 제안하는 시점 예측 기법의 구현에 대해 단계적으로 서술하고, 실험 결과를 제시한다. 마지막으로, 본 기법의 의의와 향후 연구 과제를 도출한다.

## 2. 배경연구

## 가. MPEG-DASH

HTTP 적응 스트리밍 기술은 표준 HTTP를 이용하여 클라이언트 상황에 적용할 수 있도록 콘텐츠를 초 단위의 세그먼트로 쪼개어 전송하는 기술이다. 이는 클라이언트의 네트워크 상황이 좋은 경우 코화질의 세그먼트를 전송한다. 모든 세그먼트를 전송할 때 까지 클라이언트의 네트워크 상황을 관측하여 전송 도중 네트워크의 환경이 열악하면 저화질의 세그먼트를 전송해 버퍼링으로 인한 재생중단을 막는다. MPEG(Moving Picture Experts Group)은 HTTP 적응 스트리밍 기술과 관련하여 2010년 DASH 그룹을 제정해 관련된 스트리밍 패키지 포맷에 대한 표준화를 시작하였고, 2012년 4월 국제 표준으로

MPEG-DASH를 ISO/IEC 23009-1 :2012로 게시하였다. 이는 HTTP 서버에 비디오 스트림을 해상도별로 인코딩하고 초 단위의 세그먼트로 쪼개어 이를 기술하는 문서인 MPD(Media Presentation Description) 파일과 함께 저장해둔다. 이 후, DASH 클라이언트의 다운로드 요청이 있을 시 가장 먼저 MPD를 전송하고 클라이언트는 이를 파싱(parsing) 하여 네트워크 상황 및 요구조건에 따라 세그먼트 정보를 확인 후 요청하고 받아와 끊김없이(seamless) 스트리밍 재생한다. 이 과정은 클라이언트의 요청 중지 혹은 미디어 스트림의 재생이 끝날 때까지 반복된다 [4].

**나. 시점 예측 기법 (Viewport Prediction)**

그림 1과 같이 360 VR 콘텐츠 내 사용자의 실 시정영역은 일부분에 불과하다. 그러나 360 VR 콘텐츠는 360도 공간 전체로 전송되어야 하므로 비 시정영역의 영상을 전송하는 만큼의 불필요한 요구대역폭이 발생한다. 이는 곧 대역폭이 한정적이고 가변적인 모바일 네트워크에서 초기 접속 지연 및 재생 중단과 같은 문제를 야기해 결과적으로 사용자의 만족도를 감소시키는 결과를 초래한다.

이러한 문제를 해결하기 위해 전체 요구대역폭을 감소시키는 것을 목표로 360 VR 콘텐츠 내 사용자의 시점을 예측하고 시점에 해당하는 영역에 높은 비트율을 할당하고 그 외 영역에 낮은 비트율을 할당하여 전송하는 기법들이 연구되고 있다. 이는 기본적으로 사용자의 시점에 해당하는 부분에 높은 비트율을 할당하며, 그림 2와 같이 영상 내 사용자 관심정보(Region Of Interest, ROI) 혹은 확률 정보 등을 기반으로 사용자의 다음 시점을 예측하고, 시점에 해당하는 영역에 부분적으로 높은 비트율을 할당한다[5].

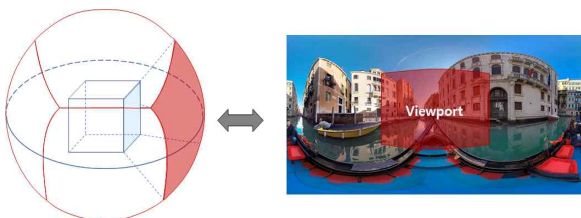


그림 1. 360 VR 콘텐츠 내 사용자의 시점(Viewport)

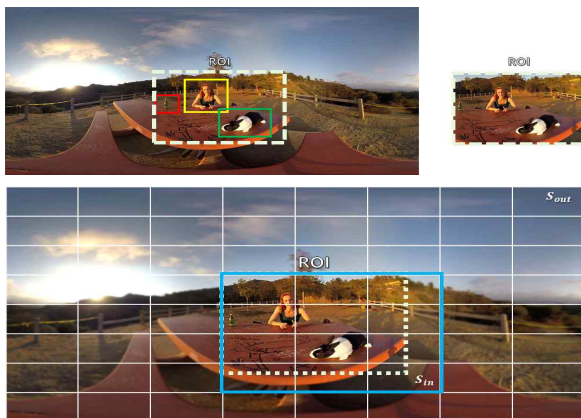


그림 2. 사용자 관심정보 기반 사용자 시점 예측 기법

**3. 음원위치정보를 활용한 시점 예측 기법**

본 논문의 전송은 MPEG-DASH 기술을 기반으로 하며 사용자 시점에 해당하는 영역에 높은 비트율을 할당하기 위해 하나의 파노라마 영상을 6개의 영상으로 나누어 독립적 병렬프로세스로 전송한다. 웹 기반 360 VR 콘텐츠 전송을 위하여 WebVR의 프레임워크인 aframe을 활용하여 360도 공간상에 지오메트리를 지정하고, 각 영상의 MPD를 스스로 연결하여 렌더링한다. 본 논문은 사용자의 오리엔테이션(i.e. Yaw, Pitch, Roll) 중 요(i.e. Yaw)값과, 각 영상의 지오메트리 값을 활용하여 각도 기반의 사용자 시점 예측 알고리즘을 제안한다.

**가. MPEG-DASH 기반 6-DASH 전송**

본 논문에서 제안하는 시점 예측 알고리즘을 기반으로 영상의 부분적 비트를 할당을 위해 먼저 하나의 파노라마 영상을 6개의 영상으로 나누어 구성한다. 원본 파노라마 영상의 프레임 너비와 높이는 각각 1920, 960이고, 이를 다음과 같이 두 개의 Pole과 네 개의 Equator로 구성하였으며 결과는 그림 3과 같다. :

- out 1(Pole) : 1920 \* 320 (0,0,1920,320)
- out 6(Pole) : 1920 \* 320 (0,640,1920,320)
- out 2(Equator) : 480 \* 320 (0,320,480,320)
- out 3(Equator) : 480 \* 320 (480,320,480,320)
- out 4(Equator) : 480 \* 320 (960,320,480,320)
- out 5(Equator) : 480 \* 320 (1440,320,480,320)

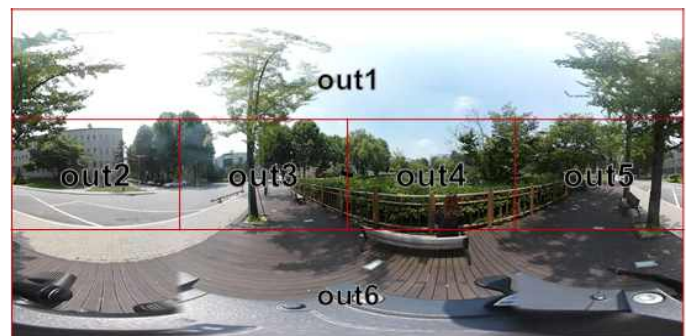


그림 3. 6 개의 영상으로 구성된 결과

각 영상에 대해 (100, 500, 1500)kbps의 해상도로 인코딩 후 3 초 단위의 세그먼트로 나누고 MPD를 생성하여 HTTP 서버에 저장한다.

**나. WebVR 프레임 워크를 활용한 Web 기반 360VR콘텐츠 전송**

이 후, 6 개의 영상을 하나의 360 VR 콘텐츠로 렌더링하기 위하여 WebVR의 프레임 워크인 aframe의 <a-sky>태그를 활용한다. <a-sky>태그에 각 영상 별 지오메트리 값을 지정하고, 스스로 <video>태그에 지정한 MPD url을 연결한다. <a-sky> 태그에 지정한 지오메트리 값은 (phi-start, phi-length, theta-start, theta-length) 값이며 그림 4와 같이 구성하였다.

```

<a-assets>
  <video id="out1" data-dashjs-player autoplay loop="true" src="/tile5/out1/out1.mpd"></video>
  <video id="out2" data-dashjs-player autoplay loop="true" src="/tile5/out2/out2.mpd"></video>
  <video id="out3" data-dashjs-player autoplay loop="true" src="/tile5/out3/out3.mpd"></video>
  <video id="out4" data-dashjs-player autoplay loop="true" src="/tile5/out4/out4.mpd"></video>
  <video id="out5" data-dashjs-player autoplay loop="true" src="/tile5/out5/out5.mpd"></video>
  <video id="out6" data-dashjs-player autoplay loop="true" src="/tile5/out6/out6.mpd"></video>
</a-assets>

<a-sky phi-start="0" phi-length="360" theta-start="0" theta-length="60" src="#out1"></a-sky>
<a-sky phi-start="0" phi-length="90" theta-start="60" theta-length="60" src="#out2"></a-sky>
<a-sky phi-start="90" phi-length="90" theta-start="60" theta-length="60" src="#out3"></a-sky>
<a-sky phi-start="180" phi-length="90" theta-start="60" theta-length="60" src="#out4"></a-sky>
<a-sky phi-start="270" phi-length="90" theta-start="60" theta-length="60" src="#out5"></a-sky>
<a-sky phi-start="0" phi-length="360" theta-start="120" theta-length="60" src="#out6"></a-sky>
    
```

그림 4. <video>태그의 정보 및 <a-sky>태그의 지오메트리 정보



그림 5. 6 개 영상의 360도 공간 상 렌더링 결과

그림 5는 6 개의 영상을 360도 공간에서 렌더링 한 결과이다. 각 영상은 독립적으로 적응적 알고리즘을 수행하며 병렬적으로 처리되므로 영상 내 부분적인 비트율 할당이 가능하다.

#### 다. 요(Yaw)와 지오메트리를 활용한 사용자 시점 예측 및 전송

사용자의 시점에 해당하는 영역에 높은 비트율을 할당하기 위해서 가장 먼저 사용자의 시점에 해당하는 영상의 정보를 획득해야 한다. 본문은 나. 에서 구성한 환경의 <a-sky>태그에 지정한 지오메트리 값과 사용자의 오리엔테이션 중 요(i.e. Yaw)값을 통해 사용자의 시점에 해당하는 영상의 정보를 획득하는 알고리즘을 제안 및 구현한다.

그림 6은 사용자의 현재 시점에 해당하는 영상의 정보를 획득하는 알고리즘의 수도 코드이다. 수도 코드의 명명은 다음과 같다. :

- $t_i$  ( $i = 1, 2, \dots, 6$ ) :  $i$  인덱스의 영상
- $y$  : 사용자의 현재 요(i.e. Yaw)
- $ps_i$  :  $i$  인덱스의 영상의 지오메트리 중 phi-center
- $pl_i$  :  $i$  인덱스의 영상의 지오메트리 중 phi-length
- $C_{out_i}$  :  $i$  인덱스의 영상의 center point
- $S_{in}$  : 사용자의 현재 시점 영역에 해당하는 영상 세트
- $S_{out}$  : 사용자의 현재 시점 외 영상 세트

먼저, 각 영상 별 center point 값  $C_{out_i}$  을 도출한다. 이는 <a-sky>태그의 지오메트리 값을 활용하여  $\text{phi-start} + \text{phi-length}/2$  로 도출하며 각도로 정의된다. 그 다음 <a-camera>태그의 orientation 값으로부터 사용자의 요(i.e. Yaw) 정보  $y$  를 획득한다. 본 알고리즘에서 카메라의 FOV(Field Of View)는 80으로 지정하였으며, 각 영상 별 지오메트리의 phi-length  $pl_i$  는 90으로 지정하였다.  $C_{out_i} - 45 < y + \text{FOV}/2 <$

#### Algorithm of Viewport Aware Bitrate Allocation

```

Input :  $t_i$  ( $i = \{1, 2, 3, 4, 5, 6\}$ ),
         $y$  (yaw value of camera),
         $ps_i$  (Phi - start),  $pl_i$  (Phi - length),
         $C_{out_i}$  (center point of  $i$  - th out tile),
         $S_{in}$  (tile set in Viewport),
         $S_{out}$  (others tile set)

Function :
1. center point of tiles
    $C_{out_i} = ps_i + pl_i/2$ 

2. Viewport aware
   if (Math.abs(y - C_{out_i}) < 90){
     //  $t_i \in S_{in}$ , others  $\in S_{out}$ 
      $t_i$ .setMinAllowedBitrateFor("video", high);
     others.setMinAllowedBitrateFor("video", low);
   }

Output : Bitrate Allocation
    
```

그림 6. 제안하는 사용자의 시점 예측 및 영상정보 획득알고리즘

$C_{out_i} + 45$  혹은  $C_{out_i} - 45 < y - \text{FOV}/2 < C_{out_i} + 45$  인 경우 즉,  $|y - C_{out_i}| < 90$  일 때 사용자의 시점 내  $C_{out_i}$  이 존재하는 것으로 판단하여 정보를 가져온다.

#### 라. dash.js API를 활용한 비트율 할당 알고리즘 구현

먼저 나. 에서 구성한 환경의 <a-sky>태그에 지정한 theta 값을 기준으로 Pole과 Equator의 사용자 시청 빈도수 실험을 진행하였다. 실험은 out1, out6의 theta 영역( $0 < \theta < 60^\circ \parallel 120^\circ < \theta < 180^\circ$ ) 내 사용자의 시점이 위치하면 Pole, 그 외 영역(i.e. out2, out3, out4, out5,  $60^\circ < \theta < 120^\circ$ ) 에 위치하면 Equator가 콘솔 창에 기록되도록 하여 빈도수를 확인하였다. 그림 7의 실험 결과를 통해 사용자의 시청 빈도수는 Equator에 비해 Pole이 현저히 낮은 것을 확인하였고, 이 실험 결과를 기반으로 Pole에 해당하는 영상(i.e. out1, out6)의 초기 시작 비트율을 상대적으로 낮게 지정하여 전체 요구 대역폭을 감소시켜 사용자의 초기 접속속도를 향상시키고자 하였다.

그 다음 dash.js API 의 setMinAllowBitrate()와 setMaxAllowBitrate()를 사용하여 다.에서 획득한 정보를 기반으로 비트율을 할당한다. 사용자의 시점에 해당하는 영상세트(i.e.  $t_i \in S_{in}$ )에는 최소 높은 비트율로 적응적 알고리즘을 수행하도록 setMinAllowBitrate("video", high)을 지정한다. 그 외 영상세트(i.e.

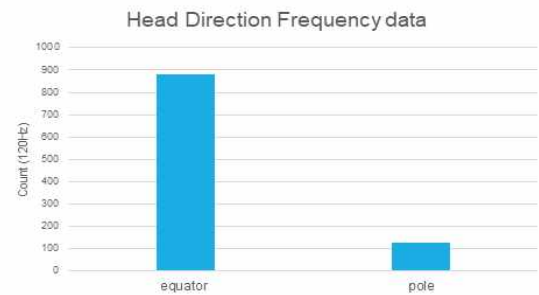


그림 7. 사용자 시청 빈도수 실험 결과

$t_i \in S_{out}$ )에는 최대 낮은 비트율로 적응적 알고리즘을 수행하도록 setMaxAllowBitrate("video", low)을 지정한다. 즉, 사용자의 시점에 해당하는 영상 세트는 최소한 높은 비트율을 보장하는 내에서 적응적 알고리즘을 수행하며 그 외 영상 세트는 최대한 낮은 비트율을 보장하는 내에서 적응적 알고리즘을 수행하도록 한다.

#### 4. 실험 결과

##### 가. 전체 요구 대역폭 (Required Bandwidth)

전체 요구 대역폭은 Chrome Dev Tools 의 네트워크 에뮬레이터를 통해 확인할 수 있다. 본 실험에서는 동일한 네트워크 환경에서 DASH 기반 파노라마 영상 전송, 본 알고리즘을 적용한 6-DASH 기반 전송의 전체 요구 대역폭을 비교하였으며 결과는 그림 8과 같다.

DASH 기반 파노라마 영상을 전송한 경우 총 18.3MB 의 대역폭을 소모하였고, 본 알고리즘을 적용한 6-DASH 기반 전송의 경우 총 27.6KB 의 대역폭을 소모한 것을 확인하였으며 결과적으로 360 VR 콘텐츠 전송 시 전체 요구대역폭의 감소를 확인하였다.

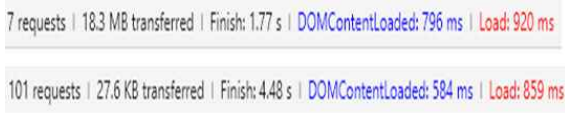


그림 8. 전체 요구대역폭 확인 결과(상: 파노라마, 하: 제안)

##### 나. 클라이언트 접속속도 (Client-Connection Speed)

클라이언트의 접속속도는 XMLHttpRequest(XHR) API를 활용한다. 먼저 XMLHttpRequest 객체를 생성한다. 그 다음 Open에 360 VR 콘텐츠를 전송하는 웹사이트의 HTML 파일을 요청하도록 설정하여 Send를 사용해 요청한다. 요청 시점의 시간을 가져와 startTime 으로 지정한다. 마지막으로 생성한 XMLHttpRequest의 객체의 readyState === 4 이고, status ===200 일 때 Request 가 완료된 시점으로 판단하여 해당 시점의 시간을 endTime 으로 지정한다. 이때, 사용자의 접속 속도는 수식 1로 계산한다[6].

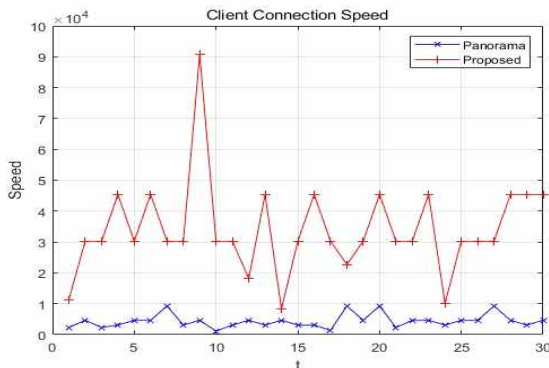


그림 9. 클라이언트 접속속도 확인 결과

$$Speed = (fileSize \times 8) / ((endTime - startTime) / 1000) / 1024$$

수식 1. 클라이언트 접속 속도 확인

본 실험에서는 동일한 네트워크 환경에서 DASH 기반 파노라마 영상 전송, 본 알고리즘을 적용한 6-DASH 기반 전송의 클라이언트 접속 속도를 1000ms 주기로 30초 간 확인 하였다. 결과는 그림 9와 같다. 실험 결과 본 알고리즘을 적용한 6-DASH 기반 전송 HTML 의 접속 속도가 빠른 것을 확인하였다.

#### 5. 결론

본 논문에서는 대역폭이 한정적이고 가변적인 모바일 네트워크의 대역폭 자원을 효율적으로 사용하는 것을 목표로 360 VR 콘텐츠의 지오메트리 값과 사용자의 요(i.e. yaw)값을 활용한 각도 기반의 사용자시점 예측 및 웹 기반 전송에 대해 연구·개발하였다. 본론에서는 제안하는 알고리즘의 구현과 전체 요구대역폭 감소 및 클라이언트 접속속도 성능의 향상을 확인하였다. 추후 본 알고리즘을 보완하여 사용자의 시점 예측의 정확도를 향상시켜 투자 자원대비 성능을 향상시킬 수 있는 알고리즘을 구현할 계획에 있다.

#### ACKNOWLEDGMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. [2016-0-00144, 시청자 이동형 자유시점 360VR 실감미디어 제공을 위한 시스템 설계 및 기반기술 연구]

#### 참고 문헌

- [1] Venturebeat, "The landscape of VR is Complicated - with234 Companies Valued at \$13B," 2015.
- [2] "HTC Vive Comes Out Top Of Sketchfab's VR Industry Trends Survey", VRfocus, 2017년 1월 11일 수정, 2018년 09월 20일 접속, <https://www.vrfocus.com/2017/01/htc-vive-comes-out-top-of-sketchfabs-vr-industry-trends-survey/>
- [3] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. Viewport-adaptive navigable 360-degree video delivery. In IEEE International Conference on Communications (ICC), 2017.
- [4] ISO/IEC 23009-1:2013, Information technology --Dynamic adaptive streaming over HTTP (DASH) -- Part1: Media presentation description and segment formats, 2013.
- [5] N. Tizon, C.Moreno, M. Preda, "ROI based video streaming for 3D remote rendering", In IEEE 13<sup>th</sup> International Workshop on Multimedia Signal Processing, China, 2011.
- [6] L.Xie, Z.Xu, Y.Ban, X.Zhang, Z.Guo, "360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming", In ACM on Multimedia Conference, 2017.