

HTML5 기반의 SHVC 비디오 플레이어용

고속 복호화 모듈 개발

www.indifrog.com

[2025. 05. 13]

발표자 : 김형근

aniri.kim@gmail.com



연구 개발 사업 개요

배경

기존 웹 환경에서 3D SHVC(Scalable High-efficiency Video Coding) 기반의 입체 영상을 재생할 수 있는 범용 플레이어는 부재

목적

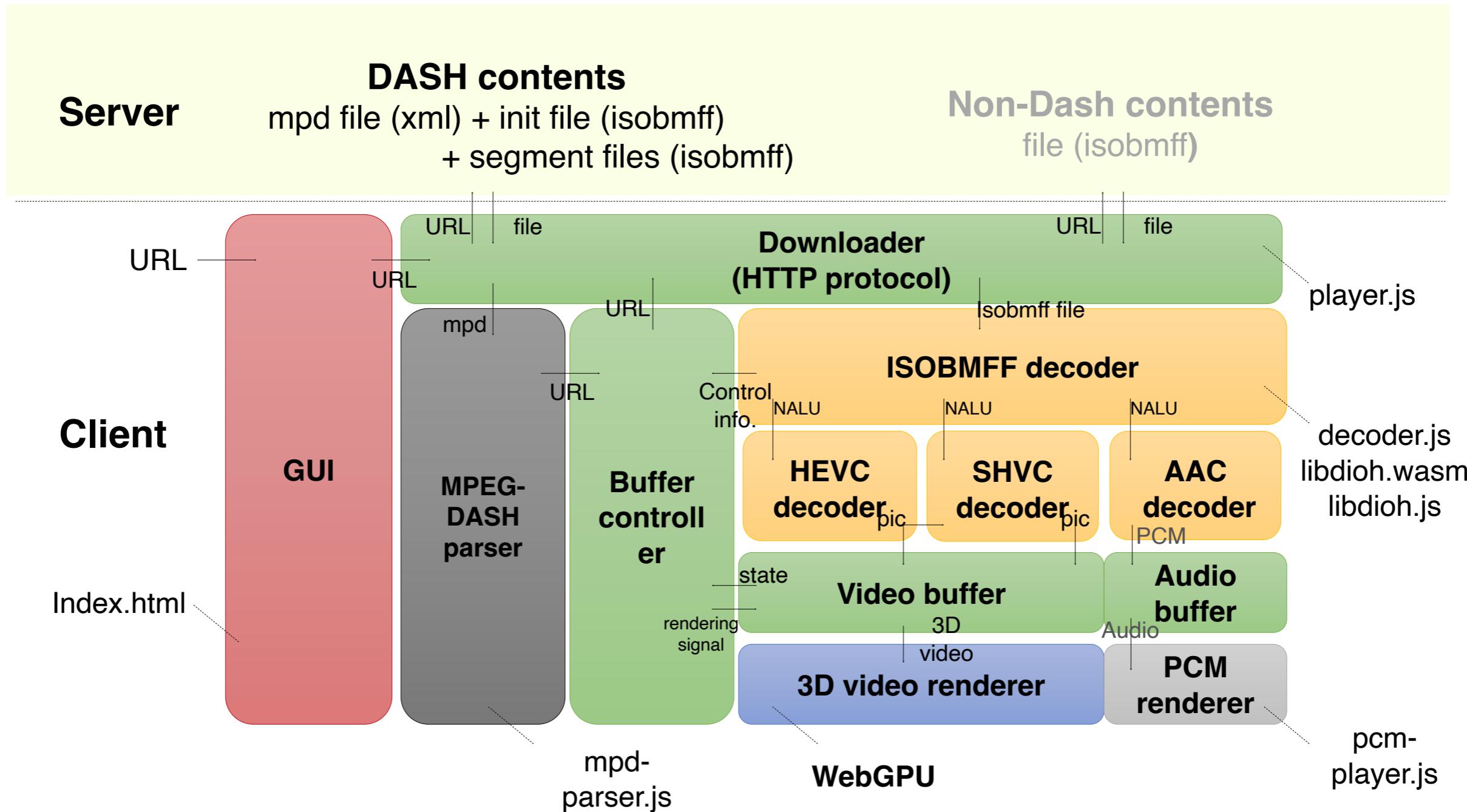
MPEG-DASH, 수정된 **FFMPEG.WASM**, HM, OpenHEVC, **WebGPU**을 활용하여 웹 브라우저 상에서 SHVC 기반 3D 입체 영상을 실시간으로 재생 및 제어 가능한 웹 플레이어 개발

기대 효과

- ❖ 몰입형 웹 콘텐츠 및 차세대 미디어 플랫폼의 핵심 기술로 활용될 수 있으며, 특히 교육, 엔터테인먼트, 원격 회의 등의 분야에서 현실감 있는 콘텐츠 소비 경험을 제공
- ❖ 표준 웹 기술 기반으로 개발되었기 때문에 플랫폼 종속성이 없으며, 다양한 디바이스 및 브라우저 환경에서 손쉽게 적용 가능
- ❖ VR/AR 기기와의 연동 확장 및 실시간 다중 사용자 콘텐츠 소비 등으로 발전 가능성



HTML5 기반 SHVC 비디오 플레이어 구조





개발 진행 과정

1. MPEG - DASH Converter

1.1 HEVC 및 SHVC 인코딩

- HM을 (SHVC 지원)를 사용 enhancement layer를 인코딩
 - Enhancement Layer: SHVC 높은 해상도 / 추가 깊이 정보
- FFMPEG (x265 지원)을 사용하여 인코딩
 - Base Layer: HEVC 낮은 해상도/품질
- HEVC, SHVC, Audio 파일로 각각 저장

DASH Converter

Video File: Choose Video File SRC_SweetRollerCoaster.mp4

Output Path: /output/dash (e.g., /output/dash/)

DASH Conversion Options

Encoding Preset: Ultrafast (Fastest) Segment Duration (seconds): 2 seconds

Audio Bitrate (kbps): 128 kbps

Video Bitrates:

- 720p 2000 kbps Remove
- 480p 1000 kbps Remove
- 360p 500 kbps Remove

+ Add Bitrate

Convert to DASH

Video Preview

0:06 / 0:29

Conversion Logs

```
[오전 7:52:58] Starting conversion of SRC_SweetRollerCoaster.mp4...
[오전 7:52:58] Output directory: /output/dash
[오전 7:52:58] File uploaded successfully to: temp/7cf46c7e-cc48-4809-b9c3-7ae5a9b133f0.mp4
[오전 7:52:58] Sending conversion request: { "inputPath": "temp/7cf46c7e-cc48-4809-b9c3-7ae5a9b133f0.mp4", "outputPath": "/output/dash/manifest.mpd", "options": { "encodingPreset": "ultrafast", "segmentDuration": "2", "videoBitrates": [ { "resolution": "1280", "bitrate": "2000" }, { "resolution": "854", "bitrate": "1000" }, { "resolution": "640", "bitrate": "500" } ], "audioBitrate": "128" } }
[오전 7:52:58] Error: Failed to analyze file structure
```

1.2. MP4 컨테이너로 래핑 (base/enh layer)

- MP4Box 등을 사용해 각 레이어를 .mp4로 래핑

1.3. MPEG-DASH 세그먼트 생성 및 MPD 생성

- MPEG-DASH 세그먼트 및 매니페스트 파일 생성

1.4 웹 서버 업로드

- MPEG-DASH 스트림과 MPD 파일을 웹 서버에 업로드
- 스트리밍 서비스



개발 진행 과정

2. MPEG - DASH Analyzer

1.1 MPEG-DASH MPD File 분석

- Stream Information
 - Type: audio, video
 - Codec: 1he1, hev1.2.4
- Segment Information
 - Duration : 90000
 - Initialization : int.mp4
 - Media : file name format

1.2. MPEG-DASH 세그먼트 분석

- MP4Box 로 MPEG-DASH 세그먼트 및 매니페스트 파일 생성
-

DASH Analyzer

Folder Selection

/uploads/sweet
Selected Path: /uploads/sweet

Apply **Browse**

DASH Analysis Results

Stream Information		Segment Information	
ID: a0		Total Segments:	0
Type:	audio	Segment Templates:	
Codec:	mp4a.40.2	Duration:	92160
Resolution:	N/A	Initialization:	3d_shvc_stream_index0_init.mp4
Frame Rate:	N/A	Media:	3d_shvc_stream_index0_\$Number\$.mp4
Bandwidth:	126 kbps	Duration:	90000
SAR:	N/A	Initialization:	3d_shvc_stream_index1_init.mp4
Scan Type:	N/A	Media:	3d_shvc_stream_index1_\$Number\$.mp4
ID: e1		Duration:	90000
Type:	video	Initialization:	3d_shvc_stream_index2_init.mp4
Codec:	lhe1	Media:	3d_shvc_stream_index2_\$Number\$.mp4
Resolution:	1280×720	Segment Files:	
Frame Rate:	24/1		
Bandwidth:	1671 kbps		
SAR:	1:1		
Scan Type:	N/A		
ID: b1			
Type:	video		
Codec:	hev1.2.4.L153.B0		
Resolution:	640×360		
Frame Rate:	24/1		
Bandwidth:	834 kbps		
SAR:	1:1		
Scan Type:	N/A		



개발 진행 과정

3. 수정된 FFmpeg.WASM을 이용한 SHVC 디코딩 및 프레임 추출

3.1 수정된 FFmpeg.WASM 초기화

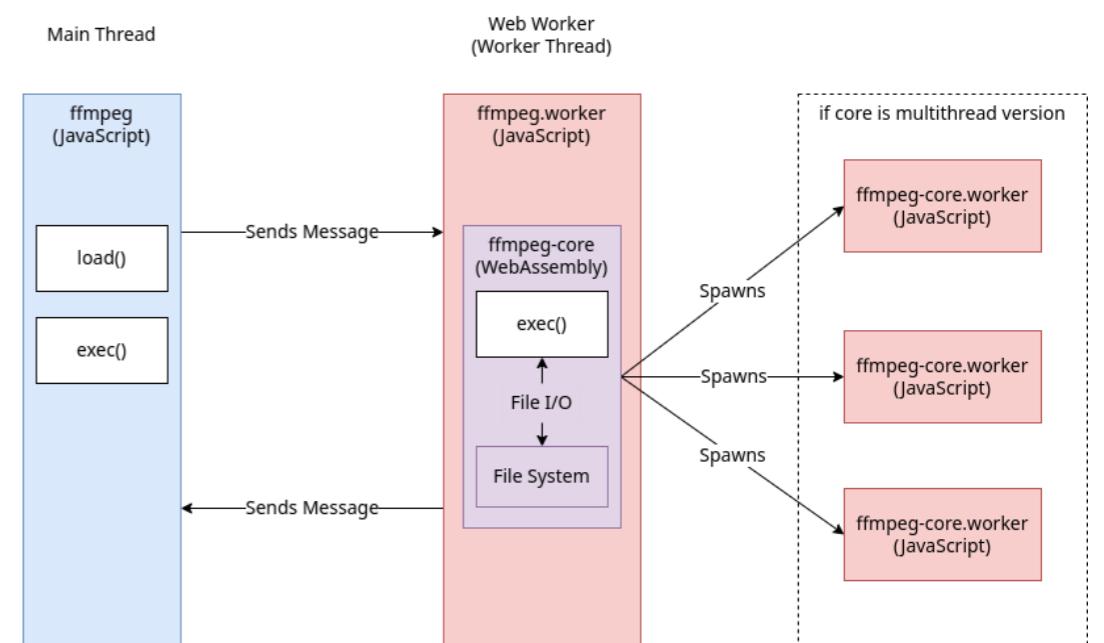
- WebAssembly 기반의 ffmpeg.wasm 모듈을 로드하고 초기화함

3.2 MPEG-DASH 스트리밍으로 수신한 SHVC 비트스트림 수집

- BL와 EL 영상을 분리해서 fetch 또는 MediaSource API로 수신
- 네트워크로 수신한 각 layer를 ArrayBuffer 형태로 메모리에 저장

3.3 SHVC 비트스트림 레이어 병합

- base + enhancement를 병합하여 하나의 프레임 영상으로 디코딩
- 좌/우 시점별 스트림 분리 디코딩 후 ImageBitmap 변환
- 디코딩 속도 향상 및 비동기 처리 최적화
- 디코딩된 프레임을 WebGPU에 렌더링 가능한 텍스처 형태로 변환





Thank You