

무안경식 사용자 선택형 입체미디어 수신모듈 구현

카이미디어

2025. 06.

전준근, 김성훈

❖ 개발 목표 및 내용

❖ 개발 현황

- 무안경식 사용자 선택형 입체미디어 수신모듈 설계
- PC 수신기용 3D 패널 인터페이스 개발
- 모바일 수신기용 3D SHVC 디코더 개발
- 테스트베드 구축 및 기능 검증

❖ Q & A

개발 목표 및 내용

❖ 개발 목표

- ATSC 3.0 및 MPEG-DASH 기반 무안경식 사용자 선택형 입체미디어 수신 모듈 개발
- 수신모듈의 입체미디어 출력과 입체미디어 전용 패널과의 정합 및 연동
- 필드테스트 수행을 통해 개발된 사용자 선택형 입체미디어 수신모듈의 정합성, 안정성, 실효성 검증

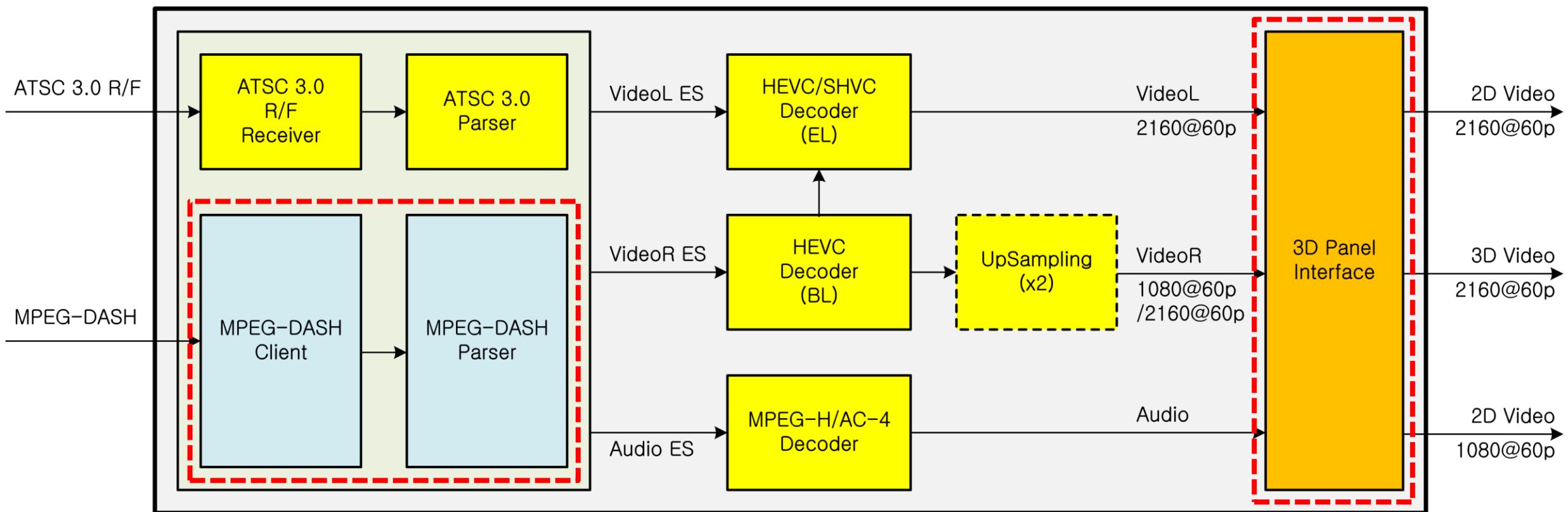
❖ 주요 내용

- ATSC 3.0 및 MPEG-DASH 기반 무안경식 사용자 선택형 입체미디어 수신모듈 개발
- 테스트베드 구축 및 기능 검증

무안경식 사용자 선택형 입체미디어 수신모듈 설계

❖ 입체미디어 수신모듈 구조

- OTA 또는 OTT 서비스 수신을 지원하기 위해 MPEG-DASH 수신 기능 추가
- 무안경식 3D 패널 인터페이스 기능 추가



PC 수신기용 3D 패널 인터페이스 개발

❖ 무안경식 3D 패널 (**MOPIC U15A731**)

- 3D 영상 전달 인터페이스 (HDMI)
- Eye tracking data 전달 인터페이스 (USB-C)



MOPIC 무안경식 3D 패널

PC 수신기용 3D 패널 인터페이스 개발

❖ 무안경식 3D 패널 규격 (MOPIC U15A731)

Display	
Screen Diagonal	396 mm (15.6")
Input Resolution	4K UHD(3840 x 2160)
Aspect Ratio	16:9
Refresh Rate	60Hz
Color Depth	16.7M
Color Space	Adobe RGB 100%
Contrast(Typ)	1500:1
Display Mode	IPS
BLU Type	Edge
3D	
3D Lens	MOPIC LF Lenticular Lens
3D lens Material	PET, Glass, UV Resin
Brightness (Typ)	320cd/m ²
Optimal Viewing Distance(OVD)	430mm
Recommended Viewing Distance	330mm~630mm
Recommended Viewing Angle	55 degree
Eye-tracking Angle	79 degree
Number of Views	stereo(2)
Crosstalk @ OVD	3.5%(@500mm)
3D Image Rate	60FPS
Eye-Tracking Rate	90FPS
3D Resolution	1.6pixel
2D/3D Switchable	No

Sensor	
Eye-tracking	Intel RealSense D435
Connection	
Video input	1 x HDMI 2.0, 2 x USB 3.1
Data interfacing	1 x USB 3.1
Audio	Audio Line Out
Speaker	2 x 2W
Power	
Type (Internal/External)	External Adaptor
Power Supply Type	PD 25W
Power Consumption (Max)	20W
Dimension	
Dimensions Without Stand	356(H) x 254.3(V) x 30.5(D) mm
Shipping Dimensions	415(H) x 280(V) x 60(D) mm
Weight	
Weight Without Stand	1.4 kg
Shipping Weight	2.2 kg
Minimum PC requirement	
CPU	Intel i7, 10th gen or better
GPU	Nvidia GeForce RTX 2080 Ti or GeForce RTX 30xx
OS Version	Windows10 1903(May 2019 update) or later
RAM	16GB dual channel or better
GPU Memory	6GB or better

PC 수신기용 3D 패널 인터페이스 개발

❖ 주요 MOPIC 3D 패널 인터페이스 SDK API

- bool **initializeEx**(const MPConfig cfg): MOPIC API 초기화
- void **finalize**(): MOPIC API 종료
- const char* **getMopicMonitorModelName**(): 모니터 모델명 리턴 (MOPIC 모니터 체크용)
- bool **startCamera**(HWND hwnd, bool show, bool enable_detector, boot is_reset): eye tracking 카메라 시작
- void **stopCamera**(): eye tracking 카메라 종료
- unsigned int **getMopicShader**(HWND hwnd, int width, int height): MOPIC 모니터에 출력할 윈도우 핸들을 제공
- void **setUse2D**(bool use): 2D/3D 모드 전환
- void **setShaderFormat**(int format): 3D 영상 형식 지정
 - 0: half side by side, 1: full side by side, 2: top & bottom

PC 수신기용 3D 패널 인터페이스 개발

“MOPIC” 버튼을 누르면, MOPIC 입체 모니터에 3D 영상 표시



무안경식 입체미디어 수신 모듈 시연 (NAB 2025)



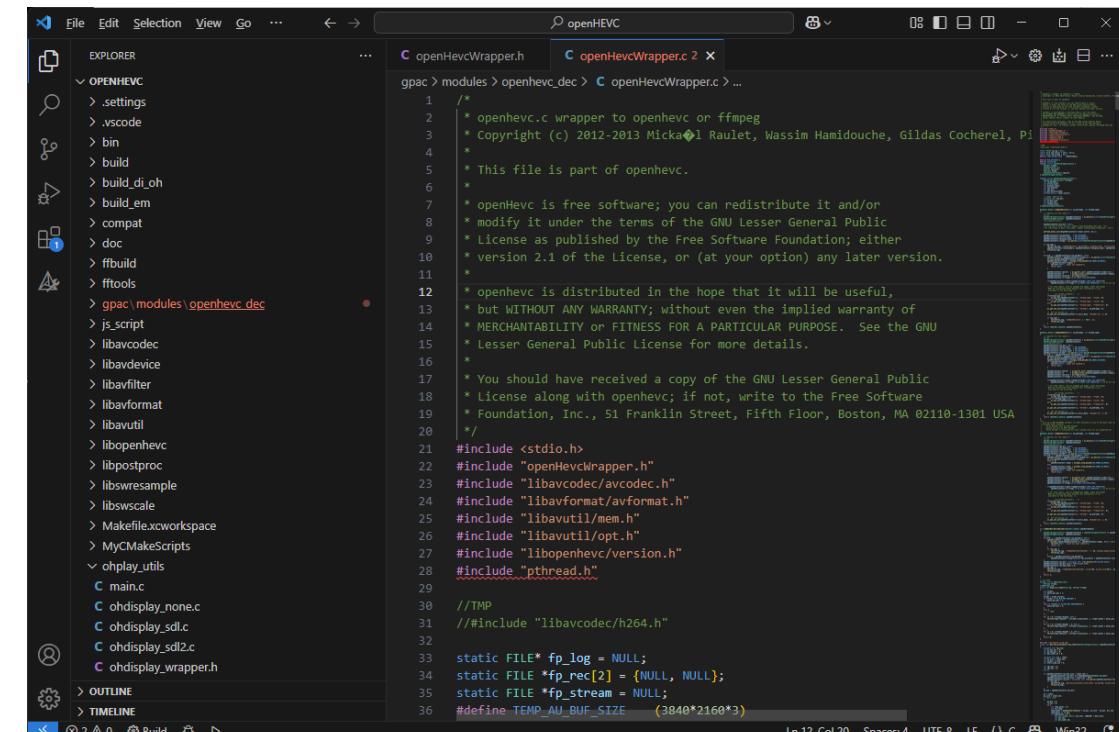
모바일 수신기용 3D SHVC 디코더 개발

❖ 모바일 수신기 환경

- H/W 디코더:
 - 대부분의 모바일 CPU가 HEVC 까지는 지원하나 SHVC는 미지원
- S/W 디코더
 - 모바일 CPU에서 실시간 동작하는 S/W SHVC 디코더 개발 필요

❖ 안드로이드용 3D SHVC 디코더 라이브러리 개발

- openHEVC 소스 기반으로 SHVC 디코딩 기능 추가된 버전 사용
- android-ndk-r28을 사용하여 빌드
- arm assembly code 사용을 enable 할 경우, 빌드 오류 발생
- C/C++ code 만을 사용하여 빌드



The screenshot shows a code editor with the following details:

- EXPLORER** pane: Shows the project structure under the **OPENHEVC** folder, including subfolders like **.settings**, **.vscode**, **bin**, **build**, **build_d1_oh**, **build_em**, **compat**, **doc**, **ffbuild**, **fftools**, **gpac** (with **modules\openhevc_dec** highlighted), **js_script**, **libavcodec**, **libavdevice**, **libavfilter**, **libavformat**, **libavutil**, **libopenhevc**, **libpostproc**, **libswresample**, **libswscale**, **Makefile.xworkspace**, **MyCMakScripts**, **ohplay_utils**, **main.c**, **ohdisplay_none.c**, **ohdisplay_sdl.c**, **ohdisplay_sdl2.c**, and **ohdisplay_wrapper.h**.
- openHvcWrapper.h** and **openHvcWrapper.c** are open in the **Code Editor** pane.
- Code Editor** pane content:

```
1  /*
2  * openhevc.c wrapper to openhevc or ffmpeg
3  * Copyright (c) 2012-2013 Micka l Raulet, Wassim Hamidouche, Gildas Cocherel, Pi
4  *
5  * This file is part of openhevc.
6  *
7  * openhevc is free software; you can redistribute it and/or
8  * modify it under the terms of the GNU Lesser General Public
9  * License as published by the Free Software Foundation; either
10 * version 2.1 of the License, or (at your option) any later version.
11 *
12 * openhevc is distributed in the hope that it will be useful,
13 * but WITHOUT ANY WARRANTY; without even the implied warranty of
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
15 * Lesser General Public License for more details.
16 *
17 * You should have received a copy of the GNU Lesser General Public
18 * License along with openhevc; if not, write to the Free Software
19 * Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
20 */
21 #include <stdio.h>
22 #include "openHvcWrapper.h"
23 #include "libavcodec/avcodec.h"
24 #include "libavformat/avformat.h"
25 #include "libavutil/mem.h"
26 #include "libavutil/opt.h"
27 #include "libopenhevc/version.h"
28 #include "pthread.h"
29
30 //TMP
31 //#include "libavcodec/h264.h"
32
33 static FILE* fp_log = NULL;
34 static FILE *fp_rec[2] = {NULL, NULL};
35 static FILE *fp_stream = NULL;
36 #define TEMP_AU_BUF_SIZE (3840*2160*3)
```
- STATUS** bar: Ln 12, Col 20, Spaces: 4, UTF-8, LF, (C), Win32

openHEVC 소스 트리

모바일 수신기용 3D SHVC 디코더 개발

❖ 안드로이드용 ffmpeg 라이브러리 빌드

- 라이브러리 구성
 - libavcodec
 - libavdevice
 - libavfilter
 - libavformat
 - libavutil
 - libpostproc
 - libswresample
 - libswscale
- ffmpeg_config.sh 파일 설정
 - arm assembly code 대신 C code로 빌드하도록 수정

```
NDK=~/android-ndk-r28
TOOLCHAIN=$NDK/toolchains/llvm/prebuilt/linux-x86_64
export PATH=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin:$PATH

chmod +x $NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android21-clang

./configure \
  --target-os=android \
  --arch=aarch64 \
  --cpu=armv8-a \
  --cross-prefix=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android21- \
  --sysroot=$NDK/toolchains/llvm/prebuilt/linux-x86_64/sysroot \
  --cc=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android21-clang \
  --cxx=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/aarch64-linux-android21-clang++ \
  --ar=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/llvm-ar \
  --ranlib=$NDK/toolchains/llvm/prebuilt/linux-x86_64/bin/llvm-ranlib \
  --enable-cross-compile \
  --prefix=./android/arm64-v8a \
  --extra-cflags="-mfloat-abi=softfp -fPIE -pie -fPIC" \
  --extra-ldflags="-fPIE -pie -fPIC" \
  --enable-static \
  --disable-everything \
  --enable-decoder=hevc \
  --enable-decoder=shvc \
  --enable-decoder=h264 \
  --disable-hwaccels \
  --disable-debug \
  --disable-vulkan \
  --enable-pthreads \
  --disable-asim
```

모바일 수신기용 3D SHVC 디코더 개발

❖ 안드로이드용 SHVC 라이브러리 빌드

- cmake에서 이용하는 CMakeLists.txt 파일을 안드로이드용 라이브러리 생성에 맞게 수정
- build_openhevc_android.sh 파일 작성 후 실행
- ~/openHEVC/build_android 폴더에 라이브러리 생성
- SHVC 디코더 라이브러리 구성
 - libLibOpenHevcWrapper.a
 - openHevcWrapper.h

```
# Android NDK 경로 설정
export ANDROID_NDK=~/android-ndk-r28
export PATH=$ANDROID_NDK/toolchains/llvm/prebuilt/linux-x86_64/bin:$PATH

# OpenHEVC 소스 디렉토리
OPENHEVC_SRC_DIR=~/openHEVC

# 빌드 디렉토리 생성
mkdir -p $OPENHEVC_SRC_DIR/build_android
cd $OPENHEVC_SRC_DIR/build_android

# CMake를 사용하여 빌드 설정
cmake -DCMAKE_TOOLCHAIN_FILE=$ANDROID_NDK/build/cmake/android.toolchain.cmake \
      -DCMAKE_SYSTEM_PROCESSOR=arm64-v8a \
      -DANDROID_ABI=arm64-v8a \
      -DANDROID_PLATFORM=android-28 \
      -DCMAKE_BUILD_TYPE=Release \
      -DCONFIG_DI=1 \
      $OPENHEVC_SRC_DIR

# 빌드 실행
cmake --build .
```

build_openhevc_android.sh

모바일 수신기용 3D SHVC 디코더 성능 측정

❖ 안드로이드용 SHVC 라이브러리 성능 측정 환경

- 테스트 기종: 삼성 갤럭시 S23+
- 영상 해상도 및 프레임율 별 성능 측정
 - 1920x1080@60p / 960x540@60p
 - 1920x1080@30p / 960x540@30p
 - 1280x720@60p / 640x360@60p
 - 1280x720@30p / 640x360@30p
 - 1280x720@24p / 640x360@24p
- 성능 측정 방법
 - 초기 성능 측정 (발열 발생 전 성능)
 - 에이징 후 성능 측정 (발열로 인한 clock down 적용)



테스트 단말: 삼성 갤럭시 S23+

안드로이드 폰 모델	싱글코어 점수	멀티코어 점수
삼성 갤럭시 S25U	2996	9531
삼성 갤럭시 S24U	2290	7376
삼성 갤럭시 S23U	2027	5152

모델별 Geekbench 점수

모바일 수신기용 3D SHVC 디코더 성능 측정

❖ 안드로이드용 SHVC 라이브러리 성능 측정 결과

영상 해상도 및 프레임율	초기 성능 (fps)	에이징 후 성능 (fps)
1920x1080@60p	77	51
1920x1080@30p	64	42
1280x720@60p	185	145
1280x720@30p	167	111
1280x720@24p	145	98

실시간 성능 총족

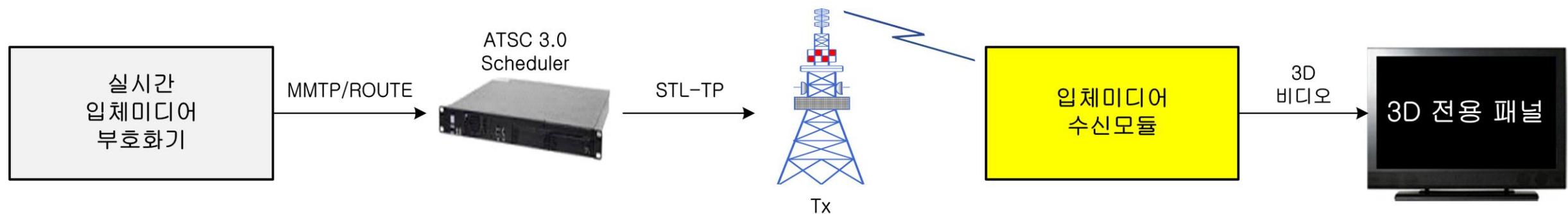
❖ 향후 개발 과제

- S/W SHVC 디코더 추가 최적화 (C code → ARM용 assembly 최적화)
- 3D 패널 인터페이스 연동
 - Unity 실시간 개발 플랫폼 연동 → MOPIC에서 제공 가능하나 DASH 수신기 통합이 어려움
 - NDK 인터페이스 → MOPIC에서 NDK API 추가 개발 필요

테스트베드 구축 및 기능 검증

❖ ATSC 3.0 기반 입체미디어 수신모듈 기능 검증용 테스트베드

- 실시간 입체미디어 부호화기
- ATSC 3.0 Scheduler
- ATSC 3.0 Transmitter
- 입체미디어 수신모듈
- 3D 전용 패널

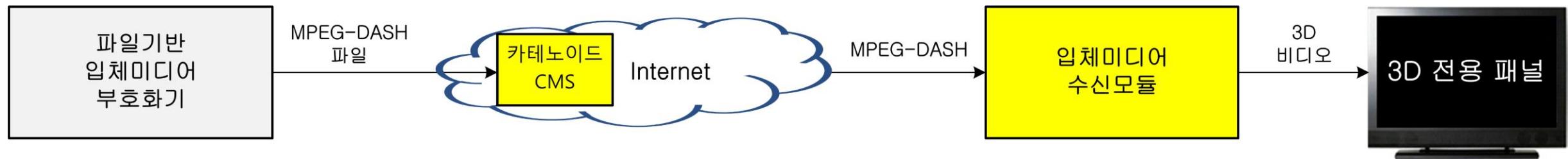


ATSC 3.0 기반 입체미디어 수신모듈 기능 검증용 테스트베드

테스트베드 구축 및 기능 검증

◆ MPEG-DASH 기반 입체미디어 수신모듈 기능 검증용 테스트베드

- 파일기반 입체미디어 부호화기
- 카테노이드 CMS
- 입체미디어 수신모듈
- 3D 전용 패널



ATSC 3.0 기반 입체미디어 수신모듈 기능 검증용 테스트베드

감사합니다