

실시간 고화질 영상에 대한 웹기반의 HLS 멀티뷰 시스템 설계

Web-based HLS(HTTP Live Streaming) Multi-view System for Real-time High Quality Video

김대진

동국대학교 영상문화콘텐츠연구원

Dae-Jin Kim(sampoo00@dongguk.edu)

요약

최근 고화질 영상 입력장치가 일반화 되고 있으며, 실시간으로 입력된 영상을 한 곳에서 동시에 볼 수 있는 중앙관제시스템이 필수 요소가 되고 있다. 이때 프로그램을 따로 설치하지 않고, 웹을 통한 접근을 하려는 시도들이 있으나, 여러 개의 고화질 영상을 동시에 웹 브라우저를 통해 시청하려하면, 웹 브라우저가 강제 종료되는 현상이 발생된다. 본 논문에서는 실시간 고화질 영상에 대한 웹기반의 HLS 멀티뷰 시스템을 제안한다. 화면으로 보이는 멀티뷰 화면을 트랜스코딩을 통해서 재구성하였고, 보안의 취약점이 있는 ActiveX를 사용하지 않으면서도, 동시에 웹 브라우저를 통해서 여러 영상 입력을 모니터링 할 수 있는 시스템을 구현 하였다.

■ 중심어 : | 웹 | HLS | 트랜스코딩 | 멀티뷰 |

Abstract

In recent years, a high-quality video input device has been generalized, and a central control system capable of simultaneously viewing video input in real time has become an essential element. At this time, there are attempts to access through the web without installing the program separately. However, if multiple high-quality videos are simultaneously viewed through the web browser, the web browser is terminated by force. In this paper, we propose a web-based HLS(HTTP Live Streaming) multi-view system for real-time high-definition video. We have reconstructed the multi-view screen as a screen by transcoding and implemented a system that can monitor multiple video inputs through a web browser on the fly without using security vulnerability ActiveX.

■ keyword : | Web | HTTP Live Streaming | Transcoding | Multi-view |

1. 서론

최근 영상 입력장치들이 FullHD(Full High Definition) 고화질 IP카메라가 널리 보급되고 있고, 4K급 이상의

고화질 콘텐츠가 IPTV나 다른 OTT 사업자에 의해서 일반화된 서비스가 되고 있으며, UHD(Ultra High Definition)카메라를 이용한 지상파 방송이 진행 중이다. IP카메라의 경우 고화질화를 통해서 감시, 보안, 상

황분석 및 제어, 교통관제, 주차 단속 등 다양한 분야에서 좀 더 정밀한 영상분석이 가능하게 되었으며, 방송 콘텐츠의 고화질화를 통해서 시청자들의 서비스 수준을 높이고 있다[1]. 특히 영상 관제 분야에서는 실시간으로 여러 영상을 한곳에서 동시에 보고, 중앙에서 상황 판단 및 대처를 하기를 원한다. 이때 웹을 통한 접근으로 관제 센터 운영요원들이 편하게 접근하기를 원하나, 고화질의 영상을 웹상에서 모니터링 하는 경우에 웹 브라우저가 강제 종료하는 문제가 발생한다. 이는 웹 브라우저가 영상 재생 용도로 만들어진 것이 아니기 때문에 고화질 영상을 처리하는데 한계가 있기 때문이다. 따라서 본 논문에서는 실시간 고화질 영상에 대한 웹기반 HLS 멀티뷰 시스템을 제안한다. 본 논문의 구성은 2장에서 기존의 단일뷰 기준의 HLS 기반의 스트리밍 시스템을 알아본다. 3장에서는 제안하는 시스템의 대한 설계를 하고, 4장에서는 구현환경과 구현 결과물에 대한 프로토타입(Prototype)을 보인 후, 기존시스템과 제안하는 시스템의 성능을 비교하였으며, 마지막으로 5장에서 결론을 지었다.

II. 기존시스템

1. 전체시스템

[그림 1]에서는 기존의 라이브 스트리밍을 위한 구성도를 나타내며, 구성에 대한 기능 설명은 아래와 같다[2].

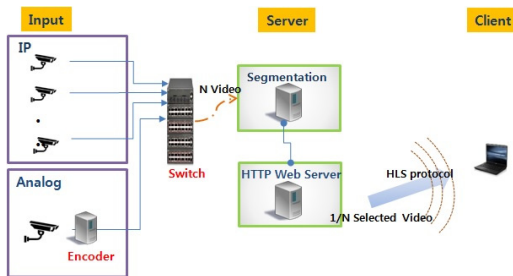


그림 1. 라이브 스트리밍 시스템 구성도

1.1 스트림 세그멘터(Stream Segmenter) 서버

IP카메라나 아날로그 카메라 등 영상장치를 통하여

다수의 영상 입력을 받는다. 입력된 영상은 RTMP(Real Time Messaging Protocol), RTSP(Real Time Streaming Protocol), HTTP(Hyper Text Transfer Protocol)등 다양한 전송프로토콜을 이용하여 스트림 세그멘터 서버로 들어온다. 스트림 세그멘터는 웹서버에서 HTTP 전송을 위해 데이터를 일정한 시간 간격으로 분할하게 된다. 분할된 데이터는 MPEG2-TS 형식으로 웹서버에 저장된다.

1.2 HTTP 스트리밍 서버

분할된 MPEG2-TS 데이터는 확장자가 m3u8인 파일에 재생 목록으로 관리된다. 영상 데이터를 재생하기를 원하는 사용자는 m3u8 파일 안에 지정된 목록대로 MPEG2-TS 데이터를 HTTP 프로토콜을 사용하여 패킷을 수신하여 플레이어 단에서 재생한다.

1.3 HLS(HTTP Live Streaming)

HLS는 Apple에서 사용하는 HTTP기반의 스트리밍 프로토콜이다. 프로토콜에서 스트리밍 데이터를 m3u8의 확장자로 재생 목록 파일과 재생 순서, 재생 시간 등 정보를 기술해 놓고 플레이어가 HTTP 프로토콜로 데이터를 수신하는 방식이다. 기존의 스트리밍 서비스 방식은 프로토콜에 맞게 다시 데이터를 변형시켜서 전송해야 하지만, HLS는 요청받은 파일을 따로 재구성하지 않고 데이터를 조각내어 전송하기 때문에 프로토콜이 비교적 간단하다. 또한 다른 스트리밍 프로토콜을 특정 포트(Port)를 열어 놓아야 되지만, HLS의 경우 추가로 포트를 열지 않고, 웹서비스를 하는 기본포트를 동시에 사용할 수 있어서 보안부분에서도 보다 안전하다.

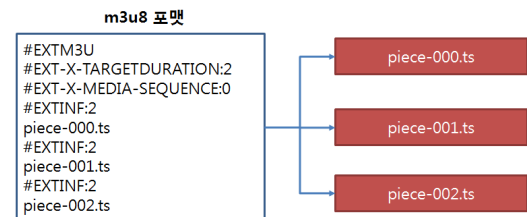


그림 2. m3u8파일과 TS 파일

[그림 2]에서는 m3u8파일과 TS 파일의 연관관계를 보여준다[3].

주요 지시어는 다음과 같다.

① #EXTM3U

파일의 맨 처음에 표시하며m3u8형식을 나타낸다.

② #EXT-X-TARGETDURATION

TS 파일들의 최대 재생시간을 나타낸다.

③ #EXT-X-MEDIA-SEQUENCE

재생해야 하는 파일의 일련번호를 나타낸다.

④ #EXTINF

재생시간과 제목을 나타낸다.

2. 문제점

기존의 라이브 스트리밍 시스템 구조는 영상 데이터를 하나씩 스트리밍하고 이를 재생하는 구조이기 때문에 웹 브라우저에서 재생할 때 정상적으로 동작한다. 그러나, 동시에 여러 영상을 모니터링하려면 제한을 받게 된다.

첫째, 브라우저 강제 종료현상이 발생한다. Internet Explorer나 Chrome 브라우저를 통해서 고화질의 영상을 동시에 여러 영상을 재생하려하면, 브라우저가 강제 종료되는 현상이 발생된다. HW 사양에 따라 죽지 않고 버티는 정도가 다르겠지만, CPU가 Intel Core i7-3630QM CPU@2.40GHz이고, 메모리가 8GB인 Windows7 64bit PC에서 FullHD급의 영상 5개를 동시

에 웹 플레이어통해서 재생할 때, 일정 시간이 지나면 죽는 현상이 발생한다. 이때 브라우저는 Internet Explorer, Chrome 모두 똑같은 현상이 발생한다. 브라우저 강제종료테스트를 10번 시도하였을 때 시점은 조금씩 다르지만, 3시간 안에 모두 강제 종료가 되는 것을 확인 하였다. 브라우저 자체가 플레이어 전용으로 만든 것이 아니기 때문에 대용량의 영상을 동시에 처리 하는데 한계가 존재하는 것으로 추정된다.

둘째, ActiveX를 이용하여 제조사 전용의 멀티뷰 플레이어를 브라우저에 삽입한다. 브라우저를 띄워 영상 입력을 재생하지만, 실제적인 영상의 관리는 제조사에서 만든 플레이어 어플리케이션이 관리하기 때문에 브라우저 종료현상이 발생되지 않는다. 단, ActiveX를 사용하기 때문에 아래와 같은 단점이 존재한다.

① Internet Explorer외 다른 브라우저에서는 사용할 수 없다. 따라서 Windows OS 외 다른 OS는 지원하지 않기 때문에, Android나 IOS 환경에서 모바일 연계가 불가능하다.

② ActiveX 설치의 문제인 의도하지 않은 프로그램 설치 때문에 보안이슈가 발생한다.

따라서, Windows 전용으로만 사용가능하기 때문에 사용자는 Windows OS 구매 비용이 별도로 들어가고, 사용자의 OS 선택권이 없다. 또한 ActiveX로 인하여 원하지 않는 파일들이 사용자 상관없이 설치되면서, 사용자가 원하지 않는 광고보기 파일이나, 다른 프로그램

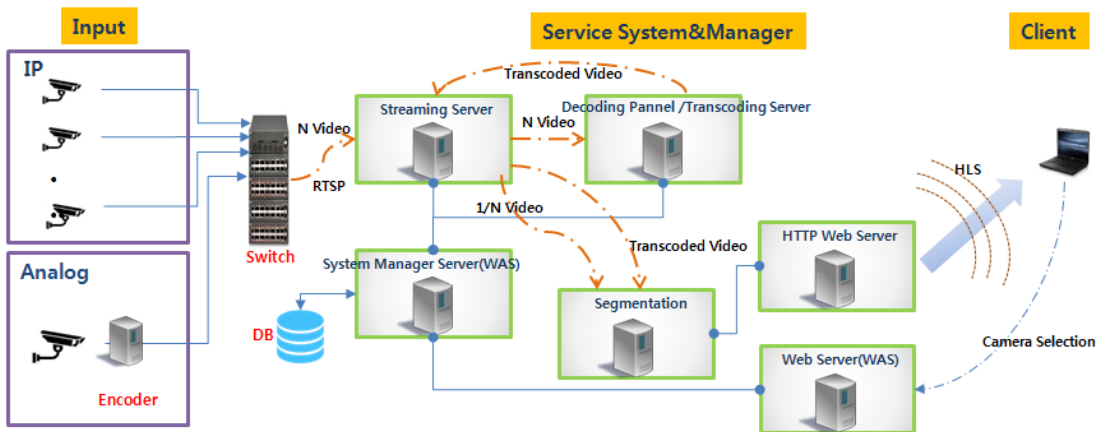


그림 3. 웹기반의 HLS 멀티뷰 시스템 구성도

에 영향을 주는 프로그램이 설치되더라도 막을 수 있는 방법이 없다.

제안하는 시스템에서는 위와 같은 문제점들을 극복하면서 여러 개의 고화질 영상을 웹기반으로 동시에 재생할 수 있는 시스템을 제안한다.

III. 제안하는 고화질 영상에 대한 웹기반의 HLS 멀티뷰 시스템

1. 전체 시스템

[그림 3]에서는 웹기반의 HLS 멀티뷰 시스템 구성도를 나타내며, 구성에 대한 기능 설명은 아래와 같다.

1.1 입력단

아날로그 카메라 또는 IP카메라로부터 영상을 캡처하여 서버단에 전달한다. 최근 IP 카메라의 경우 대부분 OnVif(Open Network Video Interface Forum)[4]를 지원하므로 RTSP로 카메라 영상을 받는다. 예전에 개발된 아날로그 카메라의 경우 Encoder를 이용하여 H.264 로 Encoding 하여 RTSP로 전달한다.

1.2 서비스 시스템 및 관리단

입력단으로부터 들어온 영상을 RTSP 프로토콜을 이용하여 사용자단에 영상을 전달한다. 이때 여러 개의 Decoding 패널을 이용하여 멀티뷰를 구성하고, 이 영상을 트랜스코딩(Transcoding)을 하여 1개의 영상 스트리밍으로 재구성한다. 이때 웹서버를 통해서 멀티뷰 뿐만 아니라 원하는 카메라 영상을 선택적으로 가져올 수 있도록 한다.

1.3 웹 플레이어(Web Player)

여러 대의 카메라로부터 전달 받은 영상 데이터를 원하는 화면을 구성할 수 있도록 선택가능하며, 웹 브라우저를 통해 멀티뷰 및 각각의 영상 데이터를 재생할 수 있다.

2. 서비스시스템 및 관리자

2.1 스트리밍서버

스트리밍 서버의 중요한 역할은 입력 받은 데이터에 대하여 여러 네트워크 환경아래에서 안정적으로 스트리밍을 하는 것이 가장 중요하다. 이와 같은 테스트 환경을 구축하기 위해서 제안하는 시스템에서는 Wowza 스트리밍 서버를 사용하였다[5]. Wowza는 RTSP, RTMP, HLS등 다양한 스트리밍 포맷을 지원한다. 스트리밍 서버의 입력은 특성에 따라 2가지로 구분할 수 있다. 첫째, 카메라로부터 직접 입력받은 영상 데이터로 IP 카메라와 아날로그 카메라 데이터를 Encoding 한 입력 데이터이다. 둘째, 이렇게 입력된 영상 데이터를 Decoding패널 및 트랜스코딩서버로 전달하여, 멀티뷰를 1개의 스트리밍 데이터로 새롭게 구성할 수 있다. 이 입력 데이터를 웹기반으로 재생할 수 있다.

2.2 Decoding 패널 및 트랜스코딩 서버

스트리밍 서버로부터 입력 받은 영상 데이터를 패널 형식으로 재구성한다. [그림 4]에서는 트랜스코딩 서버 단에서 다이렉트쇼 필터를 이용한 3x3 패널 구성 및 트랜스코딩 전송을 위한 필터 연결 구성을 보여준다[6]. 2가지 부분으로 나눌 수 있는데, 첫째, 화면에 표출하기 위한 패널파트이다. 각각의 카메라 영상을 RTSP로 받아 Demuxing하는 입력 Filter로 전달한다. RTSP 입력은 H.264로 Encoding 되어 있기 때문에 화면에 표출하기 위해서 H.264 Decoder를 통해서 YUV형태의 Raw 데이터로 디코딩을 한다. 디코딩된 Raw데이터는 EVR(Enhanced Video Render) 필터를 통해 어플리케이션 안에 있는 각 패널 번호와 맞추어서 화면에 표출된다. 둘째, 트랜스코딩파트이다. 패널안의 영상 데이터는 하나의 어플리케이션 안에서 동시에 화면 표출 된다. 결국 한 개의 어플리케이션이기 때문에 여러 개 화면은 1개의 PID를 가지게 된다. 따라서 트랜스코딩 부분에서는 PID가 차지하고 있는 좌표를 가져와서 화면 캡처를 진행한다. 그리고 멀티뷰 화면의 Raw 데이터를 H.264 Encoding을 하여 RTSP Muxer로 전달한다. 이렇게 구성된 멀티뷰 화면은 새로운 영상 스트리밍 데이터가 된다. 이 데이터는 다시 스트리밍 서버로 전송되어서 영상 감시단에서 감시할 수 있도록 한다.

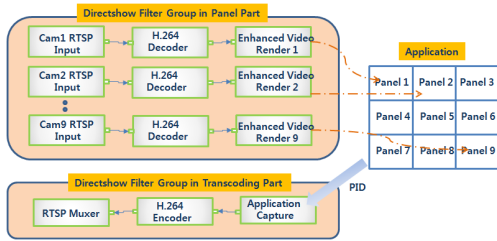


그림 4. 다이렉트쇼 필터를 이용한 3x3 패널 구성 및 트랜스코딩 전송

2.3 시스템 관리자 서버

시스템 관리자 서버에서는 2가지 역할을 한다. 첫째, 서비스 하는 각각의 서버의 서비스 가능 상태를 주기적으로 체크한다. 본 서비스에서는 1분마다 각 서버로 ping체크를 하여 서버의 어라이브(Alive) 여부를 체크하였다. 둘째, 스트리밍 서버에 연결하고자 하는 카메라 관리이다. 스트리밍 서버는 입력된 카메라의 RTSP URL 정보를 가지고 있어서 입력을 Pull형식으로 입력 받는다.

2.4 웹 서버

웹서버는 사용자가 접근할 수 있는 최고 앞단의 서비스이다. 여기서는 웹 브라우저를 통해 보고자 하는 카메라의 입력을 선택할 수 있다. 카메라 각각 1개씩 선택이 가능하고, 멀티뷰를 선택하면 여러 개의 카메라 입력도 동시에 볼 수 있다. 만약에 시스템 관리자에 등록된 카메라의 개수가 패널의 개수보다 많은 경우에는 카메라를 선택하여, 원하는 화면의 멀티뷰를 구성할 수 있다.

그밖에 스트림 세그멘터 서버와 HTTP 스트리밍 서버는 기존 시스템과 동일한 구조를 가진다.

3. 사용자

3.1 웹 플레이어

사용자가 최종으로 입력받은 영상 데이터를 볼 수 있다. HTML5 플레이어를 사용하여 프로그램 설치 없이 간단하게 사용할 수 있다.

IV. 시스템 구현결과

1. 시스템 구현환경

제안하는 시스템을 구성하기 위해서 9개의 IP 카메라를 사용하였다. 카메라로부터 입력 받은 영상은 H.264로 Encoding 되어 있으며, Main Profile환경에서 비트레이트는 6Mbps, Frame Rate는 30Frame, 해상도는 FullHD급 1080p를 사용하였다. 각각의 서버의 OS는 Windows7을 사용하였으며, 이 서버들 사양은 Intel(R) Core(TM) i7- 3630QM CPU@2.40GHz 이고, 메모리는 8GB를 사용하였다. 스트리밍 서버는 Wowza를 사용하였다. 사용자단에 스트리밍을 하기위한 HTTP 웹서버를 구성할 때 nginx를 사용하였다[7]. WAS 프로그램은 POCO 플랫폼 환경에서 VS2015 C++로 개발하였다. 코덱부분은 오픈소스인 ffmpeg을 이용하였으며, 다이렉트쇼 필터 기반으로 어플리케이션을 개발하였다. 사용자단의 플레이어는 Chrome 브라우저에서 HTML5 플레이어를 사용하였다.

2. 시스템 프로토타입

[그림 5]에서는 Decoding 패널 및 트랜스코딩 서버에서 멀티뷰를 구성하였으며, 여기에 카메라의 화면들은 각각 전송이 가능하다. 원하는 카메라 영상에 대해서 스트리밍을 할 때는 패널의 호출되는 카메라의 호출 위치좌표정보를 이용한다. 멀티뷰에 대한 트랜스코딩시 카메라 영상과 마찬가지로 Main Profile환경에서 비트

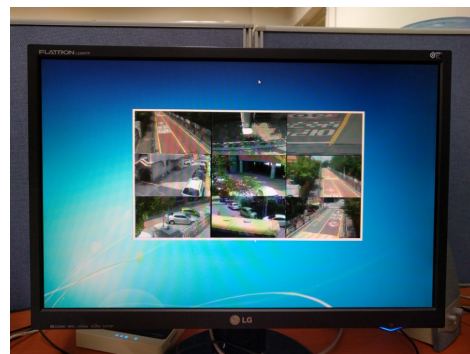


그림 5. Decoding 패널 및 트랜스코딩 서버에서 멀티뷰를 표출한 어플리케이션

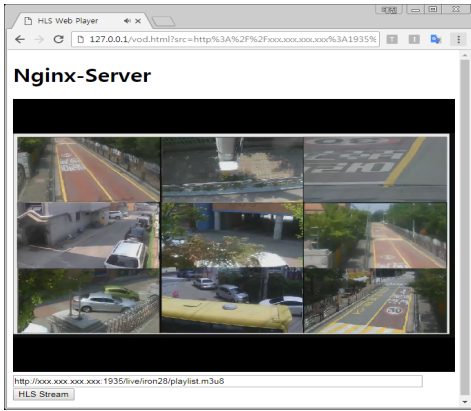


그림 6. 웹 환경에서 멀티뷰 재생

레이트는 6Mbps, 해상도는 FullHD급 1080p를 사용하였다. 재구성된 멀티뷰 화면은 [그림 6]에서처럼 Chrome 웹브라우저에서 끊김 없이 재생할 수 있음을 확인하였다.

3. 시스템 성능비교

기존 시스템은 단일 영상 스트림을 재생화면에서 확인하였다. 그러나 고화질의 영상 입력을 브라우저를 통해 동시에 여러 화면을 보는 경우에는 브라우저가 강제 종료가 발생된다. 이 부분을 해결하기 위해 ActiveX를 사용해야 되는데, 이 경우 보안 및 모바일에서 문제를 발생시킨다. 그러나, 제안하는 시스템에서는 서비스 업체의 제작 ActiveX를 사용하지 않고 HTML5 플레이어를 사용하여 별도로 플레이어를 제작하지 않고 사용하였으며, 트랜스코딩을 통해서 멀티뷰 화면을 웹기반으로 브라우저 강제종료 없이 볼 수 있다. 단, 구현시 서비스구성이 복잡해 질 수 있다. [표 1]에서는 기존시스템과 제안시스템의 특징을 비교하였다.

표 1. 기존시스템과 제안시스템의 비교 분석

항목	기존시스템	제안시스템
안정성	ActiveX 미사용시 브라우저 강제 종료발생	브라우저 강제 종료 미발생
기능성	단일뷰지원	멀티뷰지원
편리성/보안성	ActiveX사용시 보안/모바일 문제발생	HTML5 플레이어
구현 복잡성	중	상

V. 결론

실시간 고화질 영상을 웹 브라우저를 통해 동시에 여러 화면을 보게 되면 브라우저가 강제 종료되거나 보안 및 모바일에서 문제가 발생한다. 따라서 본 논문에서는 트랜스코딩을 이용하여 이와 같은 단점을 극복 할 수 있는 시스템을 제안하였다. 이를 통해 브라우저를 통해 고화질의 실시간 영상 입력을 동시에 볼 수 있다. 따라서 사용자는 접근이 편하고, 프로그램을 별도로 설치하지 않고 서비스를 이용할 수 있다. 추후 연구로, 복잡한 시스템을 단순화 시킬 수 있는 연구가 필요하다.

참고 문헌

- [1] Hiroyuki Kimiyama, Mitsuru Maruyama, Masayuki Kobayashi, Masao Sakai, and Satria Mandala, "An UHD video handling system using a scalable server over an IP network," International Journal of Advanced Media and Communication, Vol.7, No.1, pp.1-19, 2017.
- [2] 류홍남, 양길진, 김종훈, 최병욱, "IP 카메라를 위한 HTML5 기반 통합형 Live Streaming 구현," 조명전기설비학회논문지, 제28권, 제9호, pp.99-104, 2014.
- [3] R. Pantos and W. May, "HTTP Live Streaming," tools.ietf.org, Retrieved 2017.
- [4] <https://www.onvif.org>
- [5] <https://www.wowza.com/solutions/streaming-types/live-video-streaming>
- [6] 김대진, 최홍섭, "다이렉트쇼 기반의 필터차단기 설계," 한국콘텐츠학회논문지, 제9권, 제12호, pp.77-85, 2009.
- [7] <http://www.nginx.com>

저 자 소 개

김 대 진(Dae-Jin Kim)

정회원



- 1998년 2월 : 대진대학교 전자공학
학과(공학사)
- 2000년 2월 : 동국대학교 전자공
학과(공학석사)
- 2010년 2월 : 대진대학교 전자공
학과(공학박사)

- 2017년 6월 ~ 현재 : 동국대학교 영상문화콘텐츠연
구원 조교수

<관심분야> : 코덱, 멀티미디어 플랫폼, 콘텐츠 DNA,
워터마크, 딥러닝 등