

블루투스 입력의 오픈 소스기반 H.264 비디오 플레이어의 구현

박태준* · 조태훈*

*한국기술교육대학교

An Implementation of an Open Source Based H.264 Video Player for Bluetooth Input

Taejun Park* · Tai-Hoon cho*

*Korea University of Technology and Education

E-mail : crazyvirus@gmail.com

요 약

블루투스는 저전력으로 근거리상의 장치들을 무선으로 연결하여 통신하는 기술이다. 본 논문은 이러한 블루투스를 이용하여 Mpeg-4 AVC/H.264방식으로 전송하여 재생할 수 있는 프로그램의 설계와 구현에 대해서 설명한다. 본 논문에서 제안하는 동영상 플레이어는 JM기반으로 Bluetooth로 미디어 스트림을 읽어올 수 있게 기능을 추가하고 화면에 출력하기 위하여 SDL(Simple DirectMedia Layer)를 사용하였다. JM은 Mpeg-4 AVC/H.264방식의 동영상을 파일에서 읽어와 YUV420 형식의 파일로 출력을 해주는 프로그램이다.

ABSTRACT

Bluetooth is a popular wireless data transmission method with low power consumption. In this paper, we present a open source based player of video stream encoded by H.264 codec, which is transmitted via Bluetooth. An open source JM is used for H.264 decoding and modified for reading H.264 encoded data via Bluetooth. For display of decoded video data, SDL was utilized.

키워드

H.264, Multimedia, Bluetooth, JM

I. 서 론

Bluetooth[1]는 저전력 무선 통신장치로 많은 기기에서 사용되어지는 장치이다. 또한 Mpeg-4 AVC/H.264(H.264)[2] 코덱은 다른 동영상 코덱에 비하여 뛰어난 압축률을 자랑하는 동영상 코덱이다. 본 논문은 Bluetooth를 통하여 H.264 미디어 스트림을 받아 화면에 재생하는 프로그램의 설계 및 구현에 대해 기술한다.

일에서 H.264 Raw format을 읽어와 YUV420[4] 형식의 파일로 저장을 하는 프로그램이다. 이러한 JM을 이용하여 Bluetooth를 이용하여 H.264 Raw format를 전송받을 수 있도록 수정을 하고 Simple DirectMedia Layer(SDL)[5]을 이용하여 화면에 출력을 하는 프로그램을 제안한다. 그림1은 본 프로그램의 블록 다이어그램을 보여주고 있다.

II. 프로그램 구조

프로그램 구현에 있어서 사용된 Joint Model(JM)[3]은 H.264 Reference Software로써 파



그림 1. 플레이어의 블록다이어그램

그림 1에서 나온 것과 같이 Bluetooth에서 미

디어 스트림을 전송 받으면 JM을 통해 미디어 스트림을 디코딩한다. 디코딩된 영상은 YUV420형식으로 출력이 된다. YUV420형식으로 출력된 영상을 SDL라이브러리를 이용하여 화면에 출력을 한다.

III. 프로그램 구현

Bluetooth로 H.264 미디어 스트림을 전송받기 위해 미디어 스트림을 전송해주는 Server와 미리 페어링이 되어 있어야 한다. 페어링이란 Bluetooth로 통신할 기기들의 짝을 정해주는 절차로 통신을 하기 위해서는 필수적으로 해야한다. 이러한 절차를 마치고 SPP(Serial Port Profile)을 이용하여 통신을 하게 된다. SPP를 이용하면 일반적인 시리얼 포트(Serial Port) 통신과 동일하기 때문에 블루투스라고 하여 따로 특별하게 구현해야 할 부분은 없다. 기존의 JM에 있는 파일에서 읽어오는 부분을 최대한 활용하여 Bluetooth에서도 읽어올 수 있도록 구현하였다. 표1은 Bluetooth로 미디어스트림을 받기 위한 소스의 일부분이다.

표1. Bluetooth 통신부분 소스

```
// Serial Port 열기
if(!OpenPort(fn,115200))
{
    printf("Error opening port\n");
    CloseHandle(serial_port);
    exit(0);
}

// device에게 명령어 보내기
command_send(msg3);
Sleep(500);
command_send(msg2);
Sleep(500);
command_send(msg1);

// buffer 할당
annex_b->iobuffer =
    malloc (IOBUFFERSIZE * sizeof (byte));

// 할당 실패하면 프로그램 종료
if (NULL == annex_b->iobuffer)
    exit(0);

annex_b->is_eof = FALSE;
getChunk(annex_b);
```

이전에 언급한 것과 같이 Serial통신할 때와 동

일한 소스이다. OpenPort함수와 command_send 함수는 Serial통신을 하기위해 따로 구현한 함수로 overlapped I/O로 통신을 하도록 구현되어 있다. JM에서 이미 구현되어 있는 파일에서 읽어와 디코딩 하는 부분으로 넘기는 함수들을 수정하여 최대한 JM 구조가 변경되지 않도록 구현하였다.

디코딩 부분은 JM에서 구현되어 있는 소스를 그대로 사용한다. JM은 H.264 표준을 따라서 만들어진 Reference Software로 학습용으로는 유용한 프로그램이다. JM은 C로 만들어져 있으며 오픈소스이다. 대략적인 디코딩 순서를 살펴보면 그림 2와 같다.

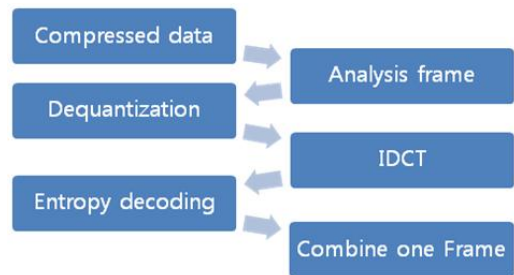


그림 2. H.264 디코딩 순서

그림 2와 같이 JM은 각 단계별로 구분되어 있고 각각 함수로 구현되어 있다.

화면 출력 부분은 SDL을 이용하여 출력을 하게 된다. SDL은 멀티미디어 라이브러리로 YUV를 다른 색공간(color space)로 변환할 필요 없이 출력해 주는 SDL_DisplayYUVOverlay 함수를 사용한다 (표 2 참조). SDL_DisplayYUVOverlay 함수는 SDL_CreateYUVOverlay를 함수를 호출 할 때 세 번째 인자로 어떤 값을 넣느냐에 따라 SDL_DisplayYUVOverlay 함수의 첫 번째 인자에 넣는 YUV값의 배열 순서가 달라진다. 본 논문은 SDL_CreateYUVOverlay를 호출할 때 세 번째 인자로 SDL_YV12_OVERLAY를 사용하였다. SDL_YV12_OVERLAY를 사용하게 되면 SDL_DisplayYUVOverlay의 첫 번째 인자인 SDL_Overlay 구조체의 pixels에 디코딩된 영상 YUV배열을 Y는 pixels[0]에 U는 pixels[2]에 V는 pixels[1]에 넣어야 한다.

표 2. SDL 함수의 원형

```
int SDL_DisplayYUVOverlay
(SDL_Overlay *overlay, SDL_Rect *dstrect);

SDL_Overlay *SDL_CreateYUVOverlay
(int width, int height, Uint32 format,
SDL_Surface *display);
```

IV. 실험 및 결과

본 논문에서 제안한 플레이어는 표3과 같은 환경에서 개발되었다.

표 3. 개발환경

CPU	AMD Phenom 9550
RAM	4GB
OS	Microsoft Windows XP SP3
IDE	Microsoft Visual Studio 2005
Network	Bluetooth 2.0 + EDR

Bluetooth를 통하여 전송한 H.264 미디어 스트림을 전송한 결과 DirectShow 기반으로 개발했던 것[6]과 같이 Bluetooth의 대역폭에 크게 영향을 받아 화질을 높이면 전송량이 많아져 화면의 변화가 많은 부분에서 느려지는 현상이 발생하였다. 따라서 화질과 딜레이는 trade off 관계로써 적절하게 조절해주는 것이 관건이었다. 그림 3에서 보는 것과 같이 전송속도의 문제를 제외한다면 원활하게 재생되는 것을 볼 수 있었다. 그림 3은 사람의 손바닥과 책의 내용을 실시간으로 촬영하여 본 논문에서 제안한 플레이어를 통하여 재생한 동영상중 일부 영상이다.



그림 3. H.264 영상 디코딩 출력 결과

V. 결 론

Bluetooth를 이용하여 JM기반으로 H.264 영상을 재생하는 플레이어를 개발하여 테스트를 해보았다. 그 결과 대역폭으로 인하여 움직임이 많은

영상에서는 화질에 대한 제약이 많았다. 이러한 단점은 Bluetooth를 사용함으로써 생기는 단점이다. JM은 Discrete Cosine Transform(DCT), Inverse Discrete Cosine Transform(IDCT)같이 연산량이 많은 연산의 최적화가 되어 있지 않다. 이러한 연산의 최적화를 하는 것이 앞으로의 연구 과제이다.

Acknowledgment

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임

참고문헌

- [1] Bluetooth SIG, Specification of the Bluetooth System, <http://www.bluetooth.com>, 2003.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard". IEEE Transactions on Circuits and System for Video Technology, vol. 13, pp. 560-575, July 2003.
- [3] JM Software, <http://iphome.hhi.de/suehring/tml>
- [4] YUV, <http://en.wikipedia.org/wiki/YUV>
- [5] Simple DirectMedia Layer, <http://www.libsdl.org>
- [6] 박태준, 조태훈 "블루투스를 이용한 DirectShow기반의 H.264 동영상 플레이어의 설계 및 구현", 한국지능시스템학회 논문지, vol 19, no 4, pp 493-498, 2009.