

Android 기반 매립환경 모바일 영상 감시시스템 구축에 관한 연구

윤연주⁰, 조성윤^{**}

*주식회사 비츠로시스

**안양대학교

e-mail:soran@vitzrosys.com*, scho@anyang.ac.kr**

A Study on Landfill Process Surveillance System based on the Android Platform

Yeon-Joo Yoon⁰, Sung-Yun Cho^{**}

*VITZROSYS Co.,Ltd.

**Dept. of Digital Media Engineering, Anyang University

● 요약 ●

생활 폐기물매립지의 효율적인 관리를 위하여 매립장 내 매립작업 진척상황과 매립작업 관리를 24시간 관리 체계를 구축하고 이를 안드로이드 기반 영상감시 시스템과 연계함으로써 시설 관리자의 상시 감시 체계를 완성하였다. 이는 매립 진척 상황과 침출수 관리를 관리사무소의 관제시스템에서만 관찰하게 되어 있어 기상변화와 위급상황에 대처능력이 전무한 실정이다. 이번 연구를 통해서 매립시설 운영자가 외부에 있는 경우에도 매립지 전경 및 침출수 상황을 관찰을 할 수 있도록 모바일 네트워크를 이용해 외부에서도 매립지 관찰을 할 수 있는 스마트폰 앱을 개발하여 24시간/365일 매립지 관찰을 할 수 있는 환경을 제공하고자 한다. 이러한 기본 개념들을 실제 안성시 생활 폐기물 매립장에 적용하기 위한 시스템 아키텍처 및 지능형 CCTV 컨트롤 시스템 스카마 등을 연구하였다.

키워드: 안드로이드(android)

I. 서론

생활 폐기물 매립시설의 지속적이고 체계적인 관리는 안전한 도시시설물 관리를 위해 필수적이다. 이를 위한 폐기물의 장기적인 감시와 매립지의 통합적인 관리시스템을 필요로 하고 있다. 폐기물매립과 관련한 체계적인 정보관리체계의 구축을 위해 매일 반복되는 폐기물의 처리과정을 통제하고 감시하기 위하여 매립 진척상황의 정보는 토지이용과 폐기물의 안정화 상황 등을 추정 하는데 반드시 필요한 정보임에도 불구하고 체계적인 관리가 이루어지지 않고 있는 실정이다. 폐기물의 매립과정과 매립 후 침출수 관리 등을 관리자의 상주 감시를 통해서 이루어지고 있으나, 야간에 관리자 퇴근 후에는 이러한 관리감독이 효율적으로 이루어질 수 없다. 또한 출입하는 차량의 체계적인 관리가 필요하고 또한 불법 폐기물이나 매립 장소의 적합성 여부를 관리하기 위한 감시시스템을 필요로 한다. 이를 위해 매립장내에서 폐기물 운반차량의 진행경로와 매립 과정, 매립위치의 파악과 매립시점의 영상정보를 기록 a 및 감시를 원격에서 24시간 진행할 수 있는 Android 모바일 환경기반 영상감시 시스템과 이를 운용하기 위한 시스템 아키텍처와 process schema를 제안하고자 한다. 이를 위해서는 이기종간에

범용적으로 사용될 수 있는 팬틸트 기능, 줌인아웃 기능이 구현될 수 있는 시스템 라이브러리가 개발 되어야 하며 모바일 환경에서 최적으로 동작될 수 있는 프로토콜도 제시 되어야 한다.

본 논문에서는 매립장내의 작업환경을 감시하고 침출수조의 환경변화를 24시간 원격에서 감시할 수 있는 애플리케이션의 시스템 모니터링 시스템을 제안하고자 한다.

II. 본론

1. Android 기반 매립지 감시 CCTV 구현 및 설계

1.1 Android 매립지 감시 CCTV 개념

Android 기반 CCTV 카메라 뷰어는 Android 장비에서 네트워크에 연결된 IP 카메라에 접속하여 라이브 영상과 음성을 재생하고 카메라의 팬틸트, 줌인아웃 동작을 제어한다. 또한 사용자에게 친숙한 안드로이드 모바일 기기의 다른 기능과 유기적으로 연동하여 언제 어디서나 쉽고 빠르게 긴급 상황에 대처할 수 있도록 도와준다. 이를 위해 원격에서 비디오 스트리밍 데이터를 수신하

여 디코딩한 후 색상 공간(color space) 변환까지 마친 후 화면에 표시하는 작업에 안드로이드 이러한 모든 기술적 배경을 OpenCORE는 안드로이드 플랫폼의 공식 멀티미디어 프레임워크이다. 미디어 재생기를 비롯한 모든 멀티미디어 관련 API는 OpenCORE 플랫폼을 기반으로 구현되어 있다. OpenCORE 플랫폼은 다양한 비디오, 오디오 코덱과 파일 형식, 스트리밍 프로토콜을 지원하며, 플러그인 방식의 아키텍처로 구성되어 있어 추가적인 코덱이나 파일 형식을 추가하는 것도 가능하다. 또한 하드웨어 가속을 지원하기 때문에 휴대폰과 같이 제한된 하드웨어 환경에서 많이 사용하고 있다. OpenCORE 플랫폼은 기본적으로 몇 가지 코덱과 형식만 기본적으로 지원하고 실제로 많이 사용하는 코덱과 프로토콜은 안드로이드 플랫폼을 장착하는 하드웨어 벤더가 선택적으로 구매해서 탑재하도록 되어 있다. 또한 OpenCORE 명세와 별도로 안드로이드 플랫폼에서는 전통적으로 CCTV 보안 카메라 시스템은 별도의 영상선으로 연결된 아날로그 카메라와 이 영상을 압축하여 디스크에 녹화하는 DVR(Digital Video Recorder) 장비와 네트워크를 통해 클라이언트 뷰어나 중앙 관제 시스템(CMS: Central Monitor System)에게 압축한 영상과 음성을 전송하는 NVT(Network Video Transmitter) 장비로 구성된다. 하지만 근래에 들어, 인터넷과 영상 압축 기술의 발달로 인해 아날로그 카메라는 대부분 IP 카메라로 대체되고 있다. IP 카메라는 영상을 직접 H.264 / MJPEG 등과 같은 압축 코덱을 이용하여 인코딩한 후 IP 네트워크를 통해 직접 클라이언트 뷰어나 중앙 관제 시스템에게 전달하는 역할을 한다.[6] 그런데, IP 카메라는, 아날로그 카메라와는 다르게, 데이터 전송에 사용하는 프로토콜 표준이 없어서 업체마다 독자적인 프로토콜을 사용함으로써 인해 응용 프로그램 간 상호 호환이 안된다는 단점이 있다. [1]-[5]

1.2 안드로이드 Native Development Kit (NDK)

안드로이드는 기본적으로 Dalvik Java Virtual machine에서 동작하는 Java application을 지원한다. Android 1.5 버전부터는 C/C++ 언어로 작성한 native 모듈도 사용할 수 있도록 NDK 개발 도구를 제공한다. 그러므로 개발자는 Java Native Interface(JNI)를 이용해 자바 어플리케이션에서 native 코드를 호출하거나, 반대로 native 코드에서 자바 객체 method를 호출할 수 있다. 다만 Android application 자체를 native 코드로만 작성하는 것은 불가능하며 반드시 Java class와 연동해서만 사용할 수 있다.

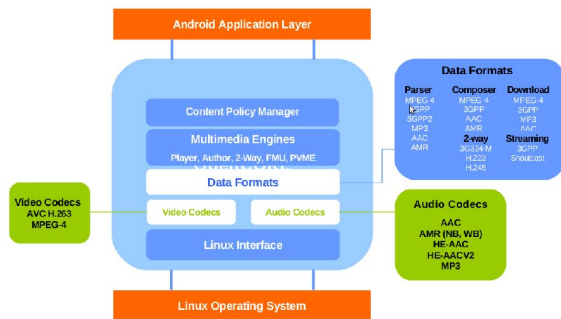


그림 1 OpenCORE 아키텍처

안드로이드 NDK는 일종의 크로스 컴파일 도구 모음이다. 즉, 개발 호스트 장비에서 타깃 플랫폼에 맞는 native 코드를 생성할 수 있도록 도와준다. 그러므로 만일 기존 C/C++ 라이브러리 소스 코드를 활용할 경우 크로스 컴파일 시 발생할 수 있는 예외사항이 모두 포함되므로 이를 고려해야 한다. 또한 자바 어플리케이션은 native 코드의 함수를 직접 호출할 수 없기 때문에 반드시 추상 자바 클래스를 만들고 그 클래스의 method를 C/C++ 언어로 하나씩 구현해야 호출이 가능하다.

1.3 Android OpenCORE C/C++ Library

안드로이드 NDK를 이용하는 C/C++ 라이브러리 목록은 표 1과 같다. FFmpeg 라이브러리는 다양한 비디오, 오디오 코덱에 대한 디코딩 기능을 제공하고, libswscale[4] 라이브러리는 YUV → RGB 변환과 이미지 확대 / 축소 기능을 제공한다. gli b/gobject [4] 라이브러리는 기본 C 라이브러리를 확장하여 객체 개념과 여러 가지 유용한 함수를 제공한다. onvif 라이브러리는 ONVIF 스펙을 만족하는 클라이언트(Network Video Client)를 구현하기 위한 관련 함수를 제공한다. edc 라이브러리는 이 모든 라이브러리를 이용하여 다양한 IP 카메라에 대한 일관된 네트워크 접속 API를 제공한다.

특히, 안드로이드(Android) 기반 ONVIF 네트워크 비디오 클라이언트는 어디서나 손쉽게 ONVIF IP 카메라에 접근할 수 있도록 도와주는 어플리케이션을 개발되어 활용가능하게 되었으며 ONVIF 스펙을 지원하는 IP 카메라 뷰어의 활용도가 더 높을 것으로 기대된다. 그러므로 본 연구의 중점 시안으로 CCTV 시스템 중 스트림 서버에서 직접 모바일 앱으로 연결될 수 있는 시스템 schema를 제안하고 이를 기반으로 원격 구동이 가능한 프로토콜을 개발하여 적용하였다.

1.4 시스템 구성

매립장 안에서 수집된 CCTV 정보를 모자일 앱으로 연계하여 영상 정보를 제공하기 위해 다음과 같이 시스템을 구성하였다. 이를 위해서 앱 어플리케이션 구조는 안드로이드 ONVIF IP 카메라 뷰어 어플리케이션의 내부 구조는 Java activity Class와 JNI Class, C 모듈 등의 세 종류 layer로 다음과 같이 구성된다[2].

자바 activity 클래스에 속하는 여러 객체는 장치 목록 관리 / 수정, 장치 스캔, 비디오 영상 표시, 오디오 재생, PTZ(팬틸트줌) 제어 등과 같이 사용자가 직접 사용하고 접하는 기능을 자바 언어로 구현한 것이다. 따라서 모든 객체는 안드로이드 SDK를 이용한다.

JNI 클래스에 속하는 Net Lib 클래스는 자바 객체이지만 자바 언어에서는 껍데기(wrapper)만 선언하고 각 method의 실제 구현(implementation)에는 안드로이드 NDK를 이용하여 C/C++로 구현되어 있다.[4] 이 객체의 API는 다시 실제 기능을 하는 C 모듈 함수를 호출한다.

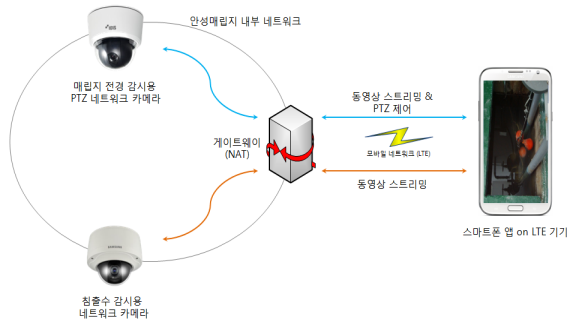


그림 2 시스템 구성도

마지막으로 C 모듈에 속하는 코드는 디코딩 / 색상 공간 변환 / 이미지 확대 / 축소 등 CPU 리소스를 많이 소모하는 작업을 처리한다.

2. 어플리케이션 제어 흐름도 및 thread 동기화

안드로이드 ONVIF IP 카메라 뷰어 어플리케이션은 사용자와 입력에 따라 네트워크에 필요한 데이터를 전송하거나, 네트워크에서 데이터가 도착했을 경우 이를 다시 사용자에게 알려주고 화면에 표시한다. 이러한 일련의 작업은 실제로 여러 thread와 layer를 거친다.

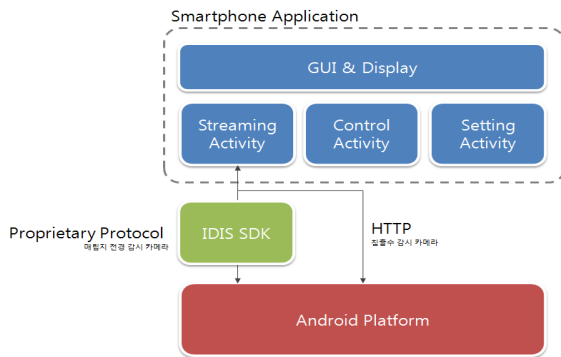


그림 3 앱 구성도

사용자가 장치에 연결하거나 스트리밍을 제어하려고 하면 제일 먼저 안드로이드 자바 activity의 해당 method가 호출된다. 이 함수는 안드로이드 GUI thread에서 호출되며, JNI 인터페이스를 통해 구현된 자바 객체 method를 다시 호출하고, 이 자바 객체 method는 다시 native C 모듈 함수를 호출한다. 하지만 네트워크를 처리하는 루틴은 안드로이드 GUI thread가 아닌 따로 생성된 native thread에서 동작하기 때문에 이 쓰레드에 메시지를 전달하려면 동기화 작업이 필요하다. 또한 반대로 native thread에서 발생한 이벤트나 프레임이 데이터는 JNI 인터페이스를 이용해 안드로이드 자바 activity 객체의 method를 호출하지만, 이 method는 다시 화면 표시 작업을 처리하는 method가 GUI thread에서 실행되도록 동기화 작업을 처리한다.

3. 실험결과

일반적으로 하나의 자바 가상머신(Java Virtual Machine) 안에는 여러 thread가 동시에 실행될 수 있는데, 이 thread 각각의 문맥(context) 정보를 포함한 객체를 실행 환경(Environment)이라고 한다. 자바 method를 이용해 thread를 생성하면 자동으로 thread별 실행 환경이 생성되지만, native C/C++ 모듈에서 생성한 thread는 실행 환경이 없다. 따라서 native 모듈에서 자바 객체를 접근하려면 먼저 자바 머신에게 thread를 알려주고, 자바 실행 환경을 만들어야 한다. 또한 각 실행 환경의 객체를 전역으로 공유할 수 있도록 요청해야 한다. 이처럼 native 모듈에서 생성한 thread와 자바 가상머신의 thread를 연동하는데 사용하는 JNI API를 사용하였다.

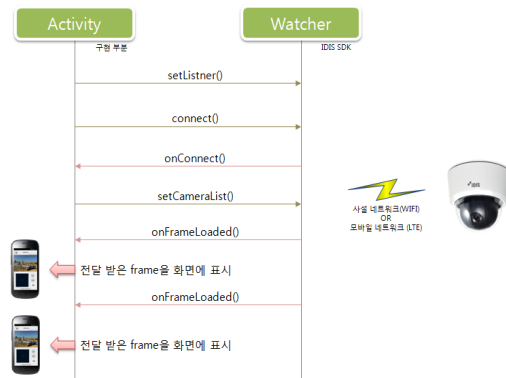


그림 4 Live view Sequence

또한 모바일 기기는 Android 기기인 GalaxyS4 LTE를 사용하였다. 동영상에서 처리할 정지 영상의 캡처를 위해

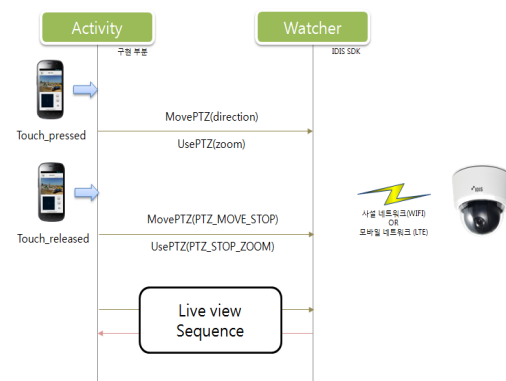


그림 5 PTZ Control Sequence

표 1 Live view syntax

<ul style="list-style-type: none"> ● HTTPS Setup <ul style="list-style-type: none"> - http://<Device IP>/cgi-bin/system.cgi?msubmenu=https&action=<value>[&<parameter>=<value>...] • Action: view [Request current information], apply[Request to apply new settings] • Connection policy ● Request JPEG <ul style="list-style-type: none"> - http://<Device IP>/cgi-bin/video.cgi?msubmenu=jpg ● Response <ul style="list-style-type: none"> - HTTP/1.0 200 OK - Content-type: image/jpeg - Content-length: <image size> - Actual image data <ul style="list-style-type: none"> • <Body> • <JPEG image data>

III. 결론

본 논문에서는 매립장내의 CCTV 영상을 Android 모바일 환경을 이용하여 원격에서 접속 제어하여 매립장내의 매립진척상황, 침출수조의 상시 감독체계를 완성하였다. 이를 위해 기본에 제공되는 Android 기반 OpenCORE 플랫폼을 적극 활용하였으며 이 기종의 IP CCTV카메라들의 효율적인 인터페이스를 위한 시스템 아키텍처, process schema, protocol들을 구현하였다. 이러한

시스템은 24시간 무중단 서비스를 모바일 환경으로 제공하여 효율적이고 신뢰성 있는 매립 관리 시스템을 제시하였다. 이로써 매립장내에서의 이루어지는 차량의 안전이동과 침출수 관리시점의 판단을 통해 적절한 폐기물의 처리과정을 감시함으로써 반입된 폐기물의 작업 효율성뿐만 아니라, 매립의 진행과 함께 작업종사자의 건강안전 고려, 환경오염 미연방지, 사후 토지 유효이용, 지반 안정화 등을 수집하고 관리하는데 이용할 수 있다.

참고문헌

- [1] ONVIF: Open Network Video Interface Forum, <http://www.onvif.org/> vol.38, Issue 4, no.13, Dec. 2006.
- [2] GLib Reference Manual, <http://library.gnome.org/devel/glib/stable/>
GObject Reference Manual, <http://library.gnome.org/devel/gobject/stable/>
- [3] Libswscale, <http://git.ffmpeg.org/p=libswscalea=summary>
- [4] GLib Reference Manual, <http://library.gnome.org/devel/glib/stable/>
- [5] GObject Reference Manual, <http://library.gnome.org/devel/gobject/stable/>
- [6] R. C. Gonzalesz and R. e. Woods, Digital Image Processing, Prentice-Hall Inc., 1998.