

# PDA에서의 MPEG-4 BIFS 파서 구현

여재욱<sup>0</sup>, 정재일, 신용경, 김상욱

경북대학교 컴퓨터학과

{jwyeou<sup>0</sup>, jijung, shinyk, swkim}@woorisol.knu.ac.kr

## Implementation of MPEG-4 BIFS Parser on PDA

Jae-Wook Yeou<sup>0</sup>, Jae-Il Jung, Yong-Kyung Shin, Sang-Wook Kim

Dept. of Computer Science, Kyungpook National University

### 요 약

본 논문은 PDA(Personal Digital Assistants)에서의 MPEG-4 BIFS(Binary Format for Scenes) 파서를 구현한다. 일반 데스크탑 환경의 MPEG-4 재생기에서 장면 구성은 BIFS 파싱 결과로 생긴 장면 그래프의 각 객체정보를 모두 렌더링 한다. 그러나, 이동 단말기인 PDA는 메모리 및 디스플레이 자원이 제한되어 있어, 기존의 데스크탑 환경에서 구현된 모듈들을 그대로 적용하기가 어렵다. 따라서, 본 논문에서는 BIFS 파싱 결과로 나온 장면 그래프에서 생성된 드로어블 노드 리스트 중 제한된 자원의 PDA에서 디스플레이 가능한 노드들만을 추출하는 새로운 기술을 제안한다. 추출된 드로어블 노드 리스트는 MPEG-4 재생기의 렌더링 모듈에서 검색하여 장면구성 및 비디오/오디오 스트림이 재생된다. 이러한 과정을 내장형 리눅스가 탑재된 PDA에서 구현하고, 기타 이동장치에서도 이용 가능하다.

### 1. 서 론

이동 환경에서는 장치의 크기가 소형화됨에 따라 메모리 및 디스플레이 등의 하드웨어 자원이 제한된다. 제한된 메모리로 인해 운영체제 및 응용 프로그램 등 소프트웨어 또한 용량이 제한되며, 크기가 작아진 디스플레이 장치를 통해 정보를 보여주기 위해서는 기존의 데스크탑과는 다른 내장형 그래픽 라이브러리가 필요하다[1]. 미디어 재생기 중 하나인 MPEG-4 재생기 구조에서 장면을 구성하는 모듈은 크게 BIFS 파서와 드로어블 노드 리스트로 나뉘어진다[2,3]. BIFS 파서는 노드 계층 정보, 노드 데이터 타입 테이블, 노드 코딩 테이블 등의 장면 정보를 이용하여 장면 그래프를 생성하고, 드로어블 노드 리스트는 생성된 장면 그래프를 렌더링 가능한 노드들의 리스트로 변환한다[4]. 하지만, 기존의 데스크탑 환경에서의 MPEG-4 재생기 기술은 데스크탑 환경에 맞게 개발되어 있기 때문에 제한된 자원의 이동 환경에 이용하는 것은 어렵다.

따라서, 본 논문에서는 제한된 자원의 PDA(Personal Digital Assistants)에서의 MPEG-4 BIFS(Binary Format for Scenes) 파서를 제안한다. 이것은 이동 환경에서 MPEG-4 미디어 재생기 개발의 일부이다. 운영체제는 내장형 리눅스를 탑재하였으며, 크로스 컴파일러를 이용하여 BIFS 파서를 컴파일 하여 PDA에 내장한다. 제안한 BIFS 파서는 장면 정보를 이

용하여 장면 그래프를 생성하는 모듈과 장면 그래프를 드로어블 노드 리스트로 변환하는 모듈을 포함하며, 드로어블 노드 리스트 구조를 생성 할 때 PDA에서 디스플레이 가능한 노드들만을 추출하도록 한다. 이러한 기술은 이동 환경에서 MPEG-4 재생기 개발에 중요하며 PDA 뿐만 아니라 휴대전화 및 기타 이동 장치에도 적용 가능하다.

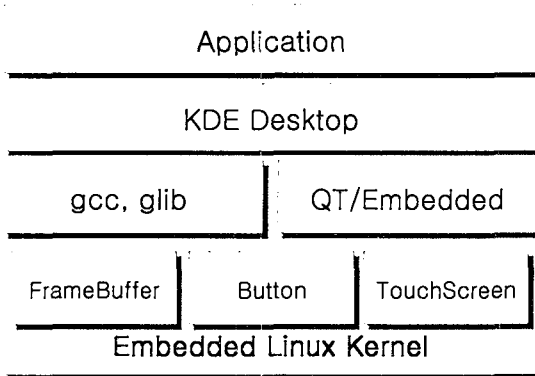
본 논문의 제 2절에서는 내장형 리눅스의 구조에 대해서 설명하고 제 3절에서는 PDA에서 구현한 BIFS 파서에 대해 설명한다. 제 4절에서는 구현모습을 보이고 마지막으로 제 5절에서 결론을 맺는다.

### 2. Embedded Linux

PDA는 StrongARM-32bit 206Mhz와 같은 RISC Processor CPU와 Touch Screen, RAM, FLASH-ROM 및 기타 입출력 장치로 구성이 된다. 여기서 FLASH-ROM은 전원이 없어도 정보의 저장이 가능하고 읽기, 쓰기가 가능한 메모리이다. 하지만, 지우는 횟수는 10만회 정도로 정해져 있다. 커널은 바로 이 FLASH-ROM에 탑재된다. 16MB의 FLASH-ROM을 0x00000000, 0x00040000, 0x00080000, 0x000c0000의 4개 영역으로 분할한 후, 각각 Boot Loader (256KB), Boot parameter (256KB), Kernel (768KB), Root file system (14.7MB)을 저장한다.

Boot loader는 FLASH-ROM에 읽기, 쓰기, 지우기, 분할 등의 관리기능을 하며, 커널을 메인 메모리에 적재하는 기능도 한다. Boot parameter는 Boot loader에 의해 관리되는 부가 정보 및 분할 정보를 저장한다. Kernel은 커널의 이미지가 저장되는 영역으로 512KB가 필요하지만, 커널 업그레이드를 위해 768KB로 설정한다. Root file system은 JFFS2 파일 시스템으로 설계되었으며 "/"로 마운트 될 파일시스템이 위치한다.

내장형 리눅스의 커널은 주로 프로세서 생성, 스레드 생성, 메모리 관리, 인터럽트 관리, 네트워크 관리 등을 지원한다. 따라서, 제한된 자원을 효율적으로 사용하기 위해 커널은 부팅 할 때 부팅에 필요한 모듈들을 먼저 적재하고, 나머지는 실행시간에 읽어들인다. 일반적인 내장형 리눅스 커널은 Loadable Module, System Type, General Setup, Memory Technology Devices, Block devices, Character devices, File systems, Console drivers로 구성된다. 내장형 리눅스 시스템은 [그림 1]과 같다.



[그림 1] 내장형 리눅스 시스템

내장형 리눅스 시스템은 커널 위에 FrameBuffer, TouchScreen, Button이 있고, 커널을 기반으로 C 컴파일러와 라이브러리, Widget 처리를 위한 QT/Embedded 불릿이 동작한다. 이러한 커널, 디바이스 및 그래픽 라이브러리를 기반으로 GUI 환경을 제공하는 응용프로그램들이 동작하게 된다. FrameBuffer는 직접 파일 입출력을 통한 그래픽 처리를 하기 때문에, 그래픽 카드에 상관없이 표준 그래픽 처리가 가능하다. 또한 디스플레이 장치와 메모리를 서로 매핑시켜 직접 하드웨어에 접근함으로써 고속의 그래픽 처리가 가능하도록 설계되었다. TouchScreen과 Button은 크기가 작은 PDA에서 사용자의 편의성을 고려한 중요한 입출력장치이다. 이러한 입

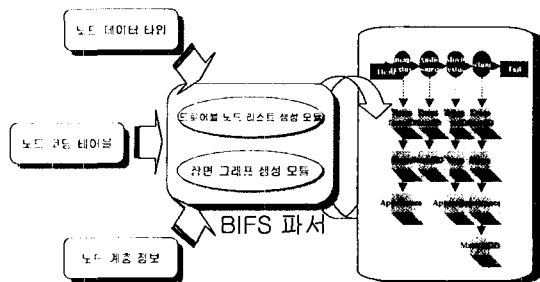
출력장치를 위한 GUI 환경을 KDE 데스크탑이 제공하고 있으며, 그 위에 MPEG-4 재생기와 같은 응용프로그램이 있게 된다.

### 3 PDA에서의 MPEG-4 BIFS 파서

기존의 MPEG-4 재생기에서의 BIFS 파서는 노드 계층 정보(NHI), 노드 데이터 타입 테이블(NDT), 노드 코딩 테이블(NCT)과 같은 장면 정보를 이용하여 하나의 장면 그래프를 생성한다. MPEG-4 재생기는 렌더링을 위해 생성된 장면 그래프를 다시 드로어블 노드 리스트로 변환해야 하며, 노드들의 속성은 서로 연관되어 구성된다. 이 모듈을 하드웨어 환경이 다른 PDA에 그대로 적용하기는 어렵다.

따라서, 본 논문에서는 기존의 BIFS 파서와 드로어블 노드 리스트 생성 모듈의 역할을 병행할 뿐만 아니라, PDA환경에 맞는 노드들만을 추출하는 새로운 방법을 제안한다. 즉, 제안한 MPEG-4 BIFS 파서는 노드 계층 정보, 노드 데이터 타입 테이블, 노드 코딩 테이블과 같은 장면 정보를 이용하여 바로 드로어블 노드 리스트를 생성하며, 생성되는 노드들은 메모리 및 디스플레이의 자원에 한계가 있는 PDA에서 프리젠테이션 가능한 노드들만 추출된 것이다.

PDA에서의 MPEG-4 BIFS 파서의 구조는 [그림 2]와 같다.



[그림 2] PDA에서의 MPEG-4 BIFS 파서

위의 그림은 제안한 BIFS 파서가 노드 계층 정보, 노드 데이터 타입 테이블, 노드 코딩 테이블과 같은 장면 정보를 이용하여 장면을 형성할 수 있는 계층 구조를 내부적으로 형성하여, PDA 환경에 프리젠테이션 가능한 노드들만을 추출하는 과정을 나타낸다. 즉, 이것은 장면 정보를 이용하여 PDA 환경에서 프리젠테이션 가능한 드로어블 노드 리스트를 생성하는 것이다. 따라서, MPEG-4 재생기는 생성된 드로어블 노드 리스트를 이용

해 속성을 가진 각 노드들을 화면에 디스플레이 할 수 있다.

디스플레이는 기존의 X-Window를 기본으로 하는 리눅스 시스템이 아니라 메모리에서 직접 LCD를 제어 할 수 있는 FrameBuffer를 사용한다. 따라서 각 모듈들은 FrameBuffer를 지원하는 내장형 라이브러리인 QT/Embedded 툴킷[6]을 이용하여 개발된다. 생성된 모듈들은 크로스 컴파일러를 이용하여 컴파일 한 후 PDA에 내장 가능하며, 터미널을 이용해서 원격에서 실행하거나 직접 PDA에서도 실행 가능하다.

#### 4. 구현

본 논문에서 사용한 내장형 리눅스 커널은 Kernel 2.4.5-rmk6-np1 이다. C와 C++의 크로스 컴파일러는 cross-armv4l-gcc-2.95.2, cross-arm4l-gcc-c++-2.95.2이며 라이브러리는 cross-armv4l-glibc-2.2.1을 사용한다. FrameBuffer를 사용하기 위한 내장형 라이브러리는 cross-armv4l-qt-2.3.1, cross-armv4l-qt\_custom-1.1, qtE-2.3.1, qtE\_custom-1.1 개발 인터페이스를 사용하였다.

PDA의 개발환경은 다음과 같다.

- 206Mhz Intel StrongARM 32bit RISC Processor
- RAM 64MB / ROM 16MB (FlashROM)
- 240 × 320 해상도의 TFT 액정, 4096 Color, 0.24mm Dot Pitch
- USB, IrDA 1.0 (115.2 Kbps Max)
- Flashing Alarm Indicator
- Speaker, Stereo Earphone Jack (3.5mm), Microphone



[그림 3] BIFS 파서 구현

내장형 리눅스 운영체제인 Linu@이 탑재된 iPAQ pocketPC 3660 PDA에 제안한 BIFS 파서를 내장시킨다 [5]. 이때, FrameBuffer를 디스플레이 장치로 사용하기 위해 QT/Embedded 툴킷에서 제공되는 그래픽 라이브러리를 사용하여 디스플레이부분을 구현한다. 내장된 BIFS 파서는 데스크탑에서 크로스 컴파일러를 이용하여 컴파일 한 후 PDA로 전송하여 직접 PDA에서 실행하였다. BIFS 파서를 구현한 화면은 [그림 3]과 같다.

#### 5. 결론

본 논문은 PDA 환경에서 장면 정보를 이용하여 장면 그래프 및 드로어블 노드 리스트를 생성하는 BIFS 파서를 구현한다. PDA는 메모리 및 디스플레이 등의 자원이 제한되어 있기 때문에, MPEG-4 파일을 재생하는데 중요한 모듈인 BIFS 파서와 드로어블 노드 리스트 생성 모듈을 하나로 통합하였다. 또한, 드로어블 노드 리스트를 생성할 때 PDA의 제한된 자원에서 재생 가능한 노드들만을 추출하도록 하였다.

향후 연구방향으로는 추출된 드로어블 노드 리스트를 이용하여 PDA 화면에 시청각 장면으로 렌더링하고, 휴대전화 및 기타 이동 장치에 적용하는 것이다.

#### 참고문헌

- [1] 박철우, "개인 휴대 정보 단말기를 위한 내장형 리눅스 설계 및 구현", 한국멀티미디어학회지 제4권 2호, pp.900-904, 2001.11.
- [2] P.J.L. van Beek, A.M. tekalp and A. Puri, "2D mesh geometry and motion compression for efficient object-based video compression," IEEE Int. Conf. On Image Processing, October 1997
- [3] L. Kovacs, "The GroupSPACE Concept," Proceedings of the Conference on Distribute Computing System, 1994.
- [4] 권순동, "동적 장면 구성을 위한 스트림 파서", 경북대학교 이학석사학위논문, pp.24-36, 1999.12.
- [5] www.linuette.com
- [6] www.trolltech.com