
안드로이드 폰에서 Jpeg Sequence 출력 시스템

강현민* · 조태훈**

한국기술교육대학교

Jpeg Sequence display system in Android

Hyunmin Kang* · Tai-Hoon Cho**

Korea University of Technology and Education

E-mail : idloveu0422@gmail.com*, thcho@kut.ac.kr**

요 약

최근 Apple의 iPhone과 Google의 Android폰 출시로 스마트 폰에 대한 관심이 증가하고 있다. 그에 따라 스마트 폰에서 사용할 수 있는 애플리케이션의 종류도 다양해지고 그로인해 스마트 폰에서 멀티미디어 콘텐츠의 활용도 증가 하고 있다. 이미지와 동영상 콘텐츠를 위해 사용되는 대표적인 코덱으로 JPEG과 MPEG 그리고 최근에 동영상 코덱으로 각광 받고 있는 H.264가 있다. 본 논문에서는 Android에서 Jpeg Sequence를 각각의 Jpeg로 분할하고 시간차를 두고 출력함으로써 동영상처럼 보이도록 한다. 그리고 이를 실제 구현을 통해서 시뮬레이션 한 결과를 보여준다.

ABSTRACT

Recently, there is increasing concern about smart phone due to the release of iPhone and Android phone. So, There are many kinds of application in smart phone, and applications using multimedia contents is increasing. The famous codecs for Image and video are JPEG, MPEG and h.264, which is popular recently as a vedeo codec because of its high compression ratio compared with the other video codecs. In this paper, we will divide Jpeg Sequence to each Jpeg and will display that Jpeg looks like a video continuously in Android. And we will show you the simulation result.

키워드

Jpeg, JpegSequence, Android

I. 서 론

최근 iPhone과 Android폰이 출시함에 따라 사용자들의 관심도 피쳐 폰(Feature Phone)에서 스마트 폰(Smart Phone)으로 증가 하고 있다. 그리고 스마트 폰에서 이미지, 동영상 등의 멀티 미디어 콘텐츠의 사용과 요구도 많아지게 되었다.

Android OS에서 멀티미디어 콘텐츠를 처리하기 위해 다양한 SDK를 제공한다. 대표적으로 Bitmap Class를 사용하면 BMP, Jpeg, PNG 등의 이미지를 시스템의 일반적인 뷰 계층구조의 드

로잉 절차에 의해 쉽게 형태와 모습을 변환하여 화면에 출력할 수 있다. 또 VideoView 위젯을 사용하면 동영상도 간단히 재생할 수 있다.[1][2]

본 논문에서는 Jpeg Sequence를 이용하여 여러 장의 Jpeg을 연속적으로 출력해 동영상처럼 보임으로써 좀 더 간단하게 동영상 콘텐츠를 활용할 수 있는 방법을 제시한다.

II. Jpeg Sequence 구조

Jpeg Sequence를 화면에 출력하기 위해서 먼저 Jpeg Sequence는 규격을 다음과 같이 정의 하였다.(표 1)(그림1)

@@@	SIZE	Image Size	#	Jpeg Data
3byte	4byte	5byte	1byte	

표 1. Jpeg Sequence Header

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 16 40 40 53 49 5A 45 20 38 33 32 30 25 FF D8 FF @@@SIZE 8320#y0y
00000010 DB 00 84 00 06 04 05 05 04 06 05 05 07 07 00 .....
00000020 06 07 09 10 0A 09 09 09 09 13 0E 0E 0B 10 17 14 .....
00000030 18 17 16 14 16 15 19 1C 24 1E 19 1B 22 1B 15 16 .....
00000040 1F 2B 20 22 25 26 28 29 28 1E 2C 2F 2C 27 2F .+ "%&() (./,/'
00000050 24 27 28 27 01 07 07 07 09 08 09 12 0A 0A 12 27 $' (!.....'
00000060 1A 16 1A 27 27 27 27 27 27 27 27 27 27 27 27 27 .....
00000070 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 .....
00000080 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 .....
00000090 27 27 27 27 27 FF C0 00 11 08 00 F0 01 40 03 01 !'!''yA...s.8..
000000A0 21 00 02 11 01 03 11 01 FF DD 00 04 00 14 FF C4 !.....yY...yA
000000B0 01 A2 00 00 01 05 01 01 01 01 01 01 00 00 00 00 .e.....
000000C0 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 10 .....
000000D0 00 02 01 03 03 02 04 03 05 05 04 04 00 00 01 7D .....
000000E0 01 02 03 00 04 11 05 12 21 31 41 06 13 51 61 07 .....
000000F0 22 71 14 32 81 91 A1 09 23 42 B1 C1 15 52 D1 F0 "q.2.'.#BzA.RN
```

그림 1. Jpeg Sequence Example

Jpeg Sequence 파일을 hexa 에디터로 보면 여러 장의 Jpeg 파일이 header와 이미지 데이터로 되어 있다. Header에는 처음 시작을 알리는 '@@@' 3byte와 이미지의 size를 나타내기 위한 'SIZE'의 4byte, 그리고 실제 데이터의 사이즈가 명시 되어 있는 5byte의 Jpeg 이미지의 사이즈, 마지막으로 header의 끝과 Jpeg의 시작을 알리는 '#' 1byte로 구성이 되어 있고 '#' 뒤로 Jpeg 이미지의 사이즈만큼의 byte로 구성이 되어 있다. 이것이 Jpeg 파일 하나의 구성이고 이러한 Jpeg가 여러 장 있는 것이 Jpeg Sequence이다.

III. Jpeg 분리

Jpeg Sequence를 출력하기 위해서 먼저 Jpeg Sequence에서 각각의 Jpeg 데이터를 추출을 해야 한다.

그러기 위해서 먼저 Jpeg Sequence 파일을 1byte씩 filestream으로 읽어서 header의 '@@@'를 찾아야 한다. 그 후에는 항상 일정하게 나오므로 'SIZE'의 'E'를 찾은 다음에 그 뒤 '#'이 나올 때 까지가 Jpeg 파일의 사이즈이므로 파일 사이즈만큼 읽어서 byte 배열에 넣는다. 그리고 이 byte 배열을 Integer.parseInt를 이용하여 int형으로 변환하여 파일의 사이즈를 구한다.[3]

그런 후에 Jpeg 사이즈만큼 읽은 byte배열을 Bitmap을 이용해 읽는다. Bitmap Class같은 경우에는 이미지가 null이거나 정상적으로 decode가 되지 않을 경우에 null을 반환하게 된다. null 값이 리턴 되었을 경우 이미지의 노이즈로 인해 이미지 데이터가 실제 명시되었던 파일의 사이즈 보다 작거나 큰 경우가 발생한다. 이때는 header에 명시했던 사이즈만큼 읽은 후 이기 때문에 1byteTlr 읽어서 다시 헤더의 '@@@'를 찾

아야 한다. null이 리턴 되고 다시 header를 찾았을 때 null이 리턴되었던 파일의 사이즈가 명시되었던 파일의 사이즈보다 클 경우 2장, 작을 경우 1장의 이미지는 버리게 된다.

Jpeg Sequence 파일을 다 읽을 때까지 각각의 이미지를 큐에 넣는다. 큐에 넣게 되면 FIFO(First In First Out)이므로 나중에 큐에서 읽었을 경우에 앞에서부터 차례대로 출력이 가능하다.

IV. Jpeg 출력

Android OS에는 2D 이미지를 출력할 수 있는 방법에는 크게 2가지 방법이 있다.[4]

첫 번째, 레이아웃의 뷰 오브젝트에 그리는 방법, 두 번째, Canvas에 직접 그리는 방법이 있다. 하지만 첫 번째 방법은 동적으로 바꿀 필요가 없는 간단한 그래픽을 그릴 때 사용하는 방법이고 두 번째 방법은 애플리케이션이 규칙적으로 스스로를 다시 그려야 할 필요가 있을 때 좋은 방법이다.

Jpeg Sequence를 출력하기 위해서는 규칙적으로 화면이 바뀌기 때문에 Canvas에 이미지를 직접 그리는 방법을 선택한다.

Canvas는 이미지가 그려지는 실제 표면에 대한 하나의 인터페이스로써 사용할 모든 draw 호출을 보유한다. Bitmap은 Jpeg를 추출할 때 모두 정의 되어서 큐에 들어 있으므로 Canvas.drawBitmap(Bitmap bitmap, float left, float top, Paint paint)를 이용하여 큐에서 하나씩 읽어서 Bitmap을 Canvas로 그린다.

이때 타이머를 이용하여 일정 시간마다 그리게 하면 여러 장의 Jpeg를 이용하여 동영상처럼 볼 수가 있다.(그림 2)

V. 결론

지금까지 Android에서 Jpeg Sequence를 재생하는 방법에 대해 제시하고 실제 구현을 통해서 시뮬레이션 한 결과를 보였다.

이 논문을 통해 Jpeg을 이용하여 더욱 간단한 동영상 콘텐츠를 처리할 수 있음을 보여 주었다.

Jpeg Sequence를 1byte씩 filestream으로 읽어서 header를 찾고 이미지 데이터를 읽어오는 방법 자체에는 문제가 없었다. 하지만 헤더에 명시되어 있는 파일의 사이즈와 실제 데이터 사이즈가 다를 경우에 다시 header를 찾기 위해 1byte씩 읽어야 할 경우 파일을 읽는데 오랜 시간이 걸린다.

향후 연구 방향으로 Jpeg Sequence를 일정 단위로 읽어 메모리상에 올리고 Jpeg Sequence를



그림 4. Jpeg Sequence 출력결과

Jpeg로 분할하는 작업을 할 계획이고 IO에 대한 시간이 줄어들기 때문에 시간이 많이 줄어들 것으로 예상된다.

참고문헌

- [1] <http://developer.android.com/reference/classes.html> : 안드로이드 공식 레퍼런스 사이트
- [2] 세인콘더 로런다시, “시작하세요! 안드로이드 프로그래밍”
- [3] <http://java.sun.com/j2se/1.3/docs/api/index.html> : Java 공식 레퍼런스 사이트.
- [4] www.kansdorid.org 운영자 들폴 양정수, “안드로이드 베이직 & 클래식 안드로이드 입문서 3rd Edition”