매트릭스연산을 이용한 안드로이드 라이브배경 개발[1]

최윤빈*, 박영호**, 최성운*
*명지대학교 컴퓨터공학과
**숙명여자대학교 멀티미디어과학과
e-mail : yoonbin82@gmail.com

Developing Android Live wallpaper using matrix operations

Yoon-Bin Choi*, Young-Ho Park**, Sung-Woon Choi*
*Dept. of Computer Engineering, MyongJi University
** Dept. of Multimedia Science, Sookmyung W. University

요 약

스마트폰 사용자가 늘어나면서 기능적인 요구를 넘어 미적인 요구가 발생되고 있다. 스마트폰의 주요 운영체제인 안드로이드가 제공하는 라이브배경(Live wallpaper)은 그러한 요구를 위한 충족시키기 위한 하나의 수단이다. 하지만 제한적인 스마트폰 환경에서 동적인 움직임을 표현하기 위해서는 효율적인 방안이 필요하다. 따라서 본 논문에서는 매트릭스 연산을 통해 한 개의 이미지를 여러개의 이미지로 표현하는 방법과 동적으로 표현하는 방법을 제시하여 라이브배경 개발 시 최소한의 리소스만을 사용할 수 있도록 돕는다.

1. 서론

스마트폰 시장이 활성화되고 사용자층이 점차 두터워지면서 사용하는 화면에 대한 다양한 요구가 발생되고 있다. 스마트폰의 주요 운영체제인 안드로이드에서는 이러한 요구를 맞추기 위해 API Level 7 부터 라이브배경(Live wallpaper)[1]를 사용할 수 있도록 지원하고 있다. 라이브배경은 동적인 움직임과 사용자 상호작용이 가능한 배경화면으로써 단순한 이미지 배경에 비해 그림 1 과 같은 다양한 표현이 가능하다.

(그림 1) 안드로이드 기반 라이브배경의 예

하지만 동적인 움직임을 표현하기 위해 지속적으로 연산을 수행해야 하기 때문에 단순 이미지를 이용한 정적인 배경에 비해 에너지 소모가 크다는 단점이 있다. 또한 다양한 표현을 위해 많은 이미지 리소스를 사용하는 경우 OS 의 프로세스 점유율이 높아지기 때문에 어플리케이션을 수행하는데 지장을 초래할 수도 있다.

따라서 본 논문에서는 라이브배경 엔진을 구성하기 위한 방법에 대해 알아보고 매트릭스연산을 통해 이미지 리소스 사용을 최소화 할 수 있는 방안을 제시한다.

2. 관련 기술

본 장에서는 안드로이드 운영체제에서 라이브 배경을 구현하기 위해 필요한 기술들을 기술한다.

2.1 배경서비스 (WallpaperService)

안드로이드에서 제공하는 배경서비스는 일반적인 안드로이드 서비스와 유사하다. 배경서비스가 일반 서비스와 다른 점은 배경서비스엔진을 생성한다는 것이다.

2.2 배경서비스엔진(WallpaperService.Engine)

배경서비스의 oncreateEngine() 메소드를 통해 생성되는 배경서비스엔진의 역할은 크게 세가지로 나뉜다. 첫째는 배경화면의 생명주기를 관리하고 둘째는 배경에 사용되는 자원을 관리하는 것이며 마지막은 사용자의 이벤트를 처리하는 것이다.

2.3 SurfaceHolder 그리기

배경서비스엔진의 getSurfaceHolder() 메소드를 통해 현재 배경화면의 그리기 전용 공간(surface)을 관리하는 SurfaceHolder 객체를 얻을 수 있다. 이는 안드로이드 시스템의 뷰 계층구조의 형태로 그리기를 기다리지 않고 보조 쓰레드를 이용하여 자신만의 속도로 canvas 에 그릴 수 있도록 지원한다.

[1] 이 논문은 2011 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2011-0002707)

3. 최소의 리소스를 이용한 라이브 배경

본 장에서는 라이브 배경에 사용되는 이미지 리소스를 최소화하기 위한 방안으로 한 개의 이미지를 컬러 매트릭스를 이용하여 여러 색의 이미지로 사용하는 방안과 좌표계 회전을 통해 동적인 이미지로 사용하는 방안을 기술 한다.

3.1 컬러 매트릭스를 이용한 이미지 변환

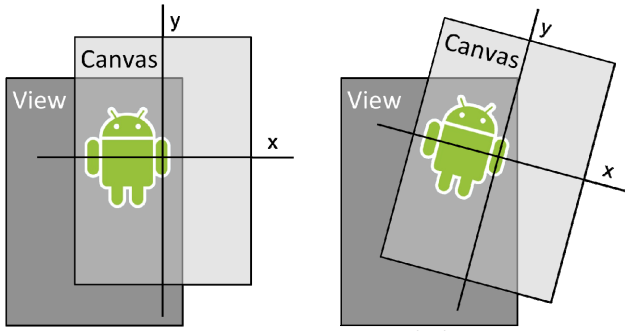
SurfaceHolder 로부터 얻은 그리기 공간(Canvas)에 이미지를 그릴 때 스타일, 색 정보를 Paint 객체에 지정하여 사용하고 있다. 이 Paint 객체에 색 필터(Color Filter)를 지정할 수 있는데 색 필터를 매트릭스를 통해 지정하게 되면 연산을 통해 새로운 값을 이미지에 적용시킬 수 있다. 이 때 사용되는 매트릭스는 5X4 형태로써 각 배열의 값을 a~t 로 지정하였을 때 그림 2 와 같은 연산을 통해 새로운 RGBA 값으로 이미지의 색을 변환한다.

[a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t]	R' = a*R + b*G + c*B + d*A + e; G' = f*R + g*G + h*B + i*A + j; B' = k*R + l*G + m*B + n*A + o; A' = p*R + q*G + r*B + s*A + t;

(그림 2) 컬러매트릭스 연산

3.2 좌표계 변환을 통한 회전효과

정적인 이미지에 동적인 효과를 주기 위한 회전효과를 사용하기 위해서는 그리기 공간(Canvas)의 좌표계를 매트릭스연산을 통해 회전해야 한다. 안드로이드의 android.graphics.Canvas 클래스는 이러한 연산을 수행하는 메소드인 rotate() 함수를 정의하여 제공한다. rotate() 함수는 회전각, 회전축을 파라미터로 받아 그리기 공간의 좌표계를 회전변환하여 그림 3 와 같이 이미지가 회전된 상태로 그려지게 한다. 이 때 그리기 공간에 그려진 기존의 그림들에 영향을 끼치지 않기 위해 save(), restore() 함수를 사용하게 된다. save() 함수를 통해 기존의 그리기공간을 스택에 저장하고 좌표계를 변환하여 그림을 그리고 restore() 함수를 통해 좌표계를 예전의 상태로 되돌린다.



(그림 3) Canvas 회전

4. 구현 및 결과

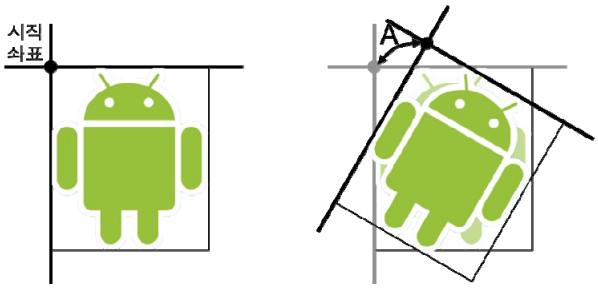
본 장에서는 매트릭스연산을 이용해 라이브배경을 구현과정을 소개한다.

4.1 멀티스레드 기반의 Refresh, Update

라이브배경은 두 가지 종류의 서브스레드로 구성되는 것이 바람직하다. 첫번째는 라이브 배경의 그리는 공간을 일정한 간격으로 새로고침(Refresh)는 스레드이고 두 번째는 눈꽃송이의 회전, 위치등을 일정한 간격으로 변경(Update)하는 스레드이다. 이 방법은 화면의 새로고침(Refresh)과 그리는 객체의 변화(Update)를 분리하는 것으로 그래픽관련 게임개발에서 주로 사용하는 방법이다.

4.2 완변한 회전을 위한 시작좌표 변화

완벽한 회전의 효과를 실현하기 위해서는 그리는 공간의 좌표계를 회전한 뒤 이미지 또한 회전각에 맞는 시작좌표를 계산하여 보정해야 한다. 시작좌표란 그림 4 에서 볼 수 있듯이 이미지의 좌측상단의 x, y 좌표를 의미한다. 그림 4 의 A 부분이 이미지 회전시 발생되는 시작좌표의 이동량이다.



(그림 4) 이미지회전에 따른 시작좌표 변화

시작좌표를 보정하는 식은 아래 식(1)과 같다. R 은 그리는 공간의 좌표계를 회전한 값이며 좌표회전 후 시작좌표를 기준으로 이미지를 그리는 공간에 그리면 완벽하게 회전된 결과를 얻을 수 있다.

$$x' = x - \cos(R/360*2\pi)$$
$$y' = y + \sin(R/360*2\pi) \quad (1)$$

5. 결론

본 논문에서는 제약사항이 많은 스마트폰 환경에서 최소한의 리소스만을 사용하여 동적인 배경화면을 개발하기 위해서 매트릭스연산을 활용하는 방안을 제시하였다. 매트릭스연산을 통해 한 개의 이미지 리소스를 다양한 색으로 변경하고 회전하면서 여러 개의 이미지 리소스를 사용하는 효과를 얻어 낼 수 있었다. 컬러매트릭스를 이용한 이미지색 변경은 라이브 배경뿐만아니라 일반 어플리케이션에서 스킨제작이나 테마 등의 효과를 사용할때도 사용될 수 있는 범용적인 방법이며 이미지회전 또한 여러 동적인 UI 를 구성하기 위해 활용될 수 있다.

참고문헌

[1] Android Developers, http://developer.android.com
[2] Android Color Matrix API, http://developer.android.com/reference/android/graphics/ColorMatrix.html