

스마트폰 중심의 N-Screen 기술

□ 권오철, 김주성, 이창건 / 서울대학교 컴퓨터공학부

요약

최근 부각되고 있는 N-Screen 기술은 서버에 저장된 콘텐츠를 PC, TV, 휴대폰을 통해 끊김 없이 이용할 수 있도록 하는 기술이다. 이와 달리, 본 고에서는 사용자가 항상 휴대하고 다니는 스마트폰이 제공하는 다양한 서비스를 사용자 주변에 있는 외부 디스플레이 장치를 활용해 보다 편리하게 제공하는 새로운 개념의 N-Screen 기술을 소개한다. 또한, 이러한 스마트폰 중심의 N-Screen 기술의 핵심인 적응형 디스플레이 전환 방법(Adaptive Display Switching Method)에 대해 소개하고, 이를 안드로이드 모바일 플랫폼에 적용한 사례를 보인다.

1. 서론

최근 서비스되고 있는 N-Screen 기술은 주로 사용자들에게 다양한 기기에 대해 하나의 영상 콘텐츠를 끊김 없이 제공하는데 그 초점이 맞추어져 있

다. 덕분에 사용자들은 가정에서 TV로 시청하던 영화나 드라마를 스마트폰이나 태블릿PC와 같은 모바일 기기를 이용해 어디에서든 이어서 감상할 수 있게 되었다. 서버상에 영상 콘텐츠와 사용자 기록들을 저장해두고 사용자가 어떠한 기기를 통해서 접속하더라도 연속적인 스트리밍 서비스를 제공할 수 있는 이러한 N-Screen 기술을 주로 One Source Multi Use(OSMU) 방식이라고 부른다. 최근에는 기기간 이어보기 기능뿐만 아니라 사용자가 TV로 시청하고 있는 영상 콘텐츠와 연관된 추가 정보 및 상품 광고 등을 스마트폰과 같은 주변기기를 통해 동시에 제공할 수 있는 Adaptive Source Multi Device(ASMD) 방식이 등장하기도 하였다.

본 고에서 제안하는 스마트폰 중심의 N-Screen 기술도 TV, PC등의 다양한 스크린을 활용한다는 점에서는 동일하다. 하지만, 스마트폰 중심의 N-Screen 기술은 영상 콘텐츠뿐만 아니라 스마트폰

을 통해 가능한 웹 브라우징, 메일 편집, 사진 감상, 페이스북, 게임 등 모든 서비스를 스마트폰 사용자 주변의 최적의 스크린 장치를 통해 제공하는 것을 목표로 한다. 이를 위해, 스마트폰이 사용자 주변에 있는 외부 디스플레이 장치를 자동으로 검색하여 연결하고, 스마트폰의 스크린 상에 보여지는 모든 이미지들을 외부 디스플레이 장치의 해상도에 맞게 변환하여 전송하는 것이 필요하다. 하지만, 현재의 모바일 플랫폼에는 이러한 N-Screen 활용을 위한 구조적 기반이 전혀 마련되어 있지 않다. 이는, 모바일 플랫폼의 구조 설계 단계부터 동적으로 스마트폰의 화면이 주변 디스플레이 장치로 전환될 수 있다는 가능성을 고려하지 않은 까닭이다. 일례로, 오픈 소스인 안드로이드 모바일 플랫폼을 살펴봐도, 디스플레이 장치에 표시될 이미지가 저장되는 프레임버퍼의 크기는 스마트폰이 가진 디스플레이 장치의 해상도에 국한되어 부팅 시부터 정적으로 고정되도록 설계되었다. 이러한 현재 모바일 플랫폼의 구조적 한계를 극복하여, 스마트폰을 보다 편

리한 인터페이스 장치로 활용할 수 있도록 하기 위한 기반 연구가 지난 2009년부터 지식경제부의 지원으로 “차세대 휴대폰용 지능형 인터페이스 플랫폼 기술개발” 과제를 통해 진행되어 왔다. 본 고에서는 스마트폰의 스크린을 사용자 주변의 외부 디스플레이 장치로 최적 전환하여 제공하는 스마트폰 중심의 N-Screen 기술에 초점을 맞추어 기술적 이슈 및 해결책을 논한다.

본 고는 다음과 같이 구성된다. 우선, 다음 장에서는 N-Screen 기술 및 디스플레이 전환과 관련된 기존 연구에 대해서 간략히 소개한다. 3장에서는 스마트폰 중심의 N-Screen 기술의 핵심 요소 기술로써, 사용자 주변의 다양한 디스플레이 장치들을 최적으로 활용할 수 있는 적응형 디스플레이 전환 방법(Adaptive Display Switching Method)에 대해 설명한다. 4장에서는 적응형 디스플레이 전환 방법을 실제 안드로이드 모바일 플랫폼 기반에 적용한 데모를 소개하고, 5장을 통해 본 고를 마무리한다.



<그림 1> N-Screen 개념의 비교

II. 관련 연구

N-Screen 기술은 서버에 저장된 콘텐츠를 PC, TV, 휴대폰을 통해 끊김 없이 이용할 수 있도록 해 주는 기술이다[2]. 이와 달리, 본 고에서 목표로 하는 N-Screen 기술은 스마트폰으로 가능한 모든 서비스들을 스마트폰 사용자의 주변에 위치한 외부 디스플레이 장치를 통해 최적으로 제공하는 것이다. <그림 1>은 통상적으로 이해되는 N-Screen의 개념과 본 고에서 목표로 하는 스마트폰 중심의 N-Screen 개념의 차이를 보인다. 그림에서 보는 바와 같이, 통상적 N-Screen은 서버에 저장된 영상 콘텐츠를 사용자의 움직임에 따라 Smart TV, PC, 휴대폰 등으로 연계해주어 사용자가 끊김 없이 영상 콘텐츠를 감상할 수 있도록 한다는 개념이다. 반면에 스마트폰 중심의 N-Screen은 영상 콘텐츠에만 국한되지 않고, 스마트폰으로 할 수 있는 모든 일들을 사용자 주변에 위치한 외부 디스플레이 장치를 통해 보다 편리하게 제공할 수 있도록 한다는 개념이다. 즉, 스마트폰이 주체가 되어 사용자 주변 기기들을 검색하고, 발견된 주변기기의 사양 정보에 따라 스마트폰의 스크린을 재구성하여 주변기기의 스크린에 표시한다. 따라서, 사용자가 소파에 앉아 TV를 보는지, 차 안에 앉아 네비게이션을 보는지, 아니면, 사무실에서 사무용 모니터를 보는지에 따라, 사용자의 주머니 안에 있는 스마트폰이 사용자의 상황에 맞게 한 개 혹은 복수 개의 화면을 생산해내고 이를 주변 디스플레이 장치에 표시하는 것이다.

스마트폰과 같은 모바일 기기의 화면을 사용자 주변의 다른 디스플레이 장치로 전환하고자 하는 시도는 다양한 방향에서 진행되어 왔다. 대표적인 방법으로는 High Definition Multimedia

Interface(HDMI)와 같은 물리적 연결을 기반으로 한 방법들이 있다. HDMI를 지원하는 스마트폰을 가진 사용자는 HDMI 케이블을 이용하여 TV와 같은 주변 장치로 스마트폰의 화면을 표시하는 것이 가능하다. 또한 최근에 규격이 마련된 WirelessHD[4] 기술이나, Intel 사가 자사 제품에 적용하기 시작한 WiDi[5] 기술을 이용하면 무선으로도 끊김 없이 Full HD급의 영상을 전송하는 것이 가능하다. 하지만 이러한 물리적 연결만을 이용한 화면 전환 방법은 단순히 화면을 확대 또는 축소하여 주변기기에 표시하는 방법일 뿐이다. 800x480 해상도의 화면을 가진 스마트폰과 1920x1080 해상도의 화면을 가진 TV를 연결하더라도 TV는 단지 800x480 해상도의 스마트폰 화면을 확대해 표시할 뿐이기 때문이다. 스마트폰과 같은 모바일 기기들이 대부분 주변기기에 비해 상대적으로 낮은 해상도를 가진다는 점에서 볼 때, 이러한 물리적 연결만을 활용한 전환 방법은 외부 디스플레이 장치들을 최적으로 활용할 수 있는 방법이 아니다.

또 다른 방법으로 Remote Desktop[6]이나 VNC[7]와 같은 응용 기반의 디스플레이 전환 방법이 있다. 이러한 응용 기반의 전환 방법을 이용하면 일부 화면의 경우 대상 디스플레이 장치의 해상도에 맞게 표시하는 것도 가능하다. 하지만 이러한 응용 기반의 방법의 한계는 스마트폰의 특정 어플리케이션에서 특정 화면에 대해서만 전환이 가능하다는 점이다.

Digital Living Network Alliance(DLNA)[8]의 규격에 따르는 제품들은 미디어 콘텐츠를 기존 홈 네트워크를 통해 서로 자유롭게 공유할 수 있다. 사용자 관점에서는 마치 스마트폰의 화면이 전환된 것으로 보일 수 있으나 내부적으로는 결국 미디어 파일을 실제 전송하여 공유하는 방식을 취하고 있

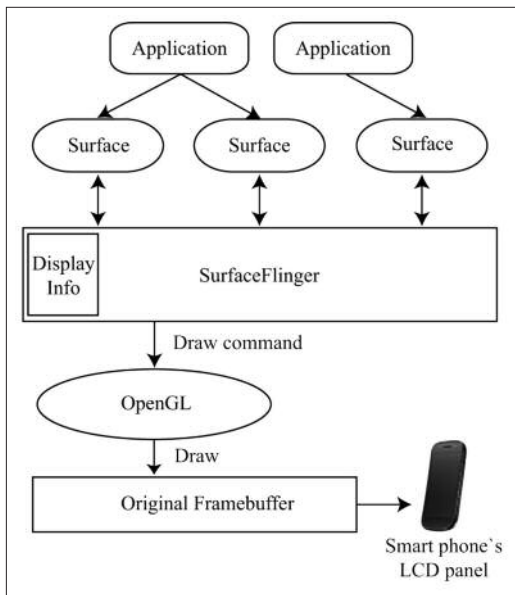
으며, 비디오, 음악, 사진과 같은 미디어 콘텐츠에 대해서만 적용이 가능하다는 제약 사항을 가지고 있다.

Motorola 사의 스마트폰인 Atrix[9]의 경우에는 랩독(Lab Dock)이라는 랩탑 형태의 도킹 장치를 이용한다. 이는 물리적 연결 기반의 방법과 응용 기반의 방법이 결합된 형태라고 볼 수 있는데, Atrix가 랩독에 연결되면 1355x768의 해상도를 가진 랩독의 화면에 맞게 해상도가 전환되고 일부 응용 또한 랩독을 통해서 이용할 수 있다. 하지만 Atrix의 경우도 응용 기반의 화면 전환 방법과 마찬가지로 모든 화면의 전환이 불가능하고, 두 장치 사이에서만 화면 전환을 지원한다는 단점을 가지고 있다.

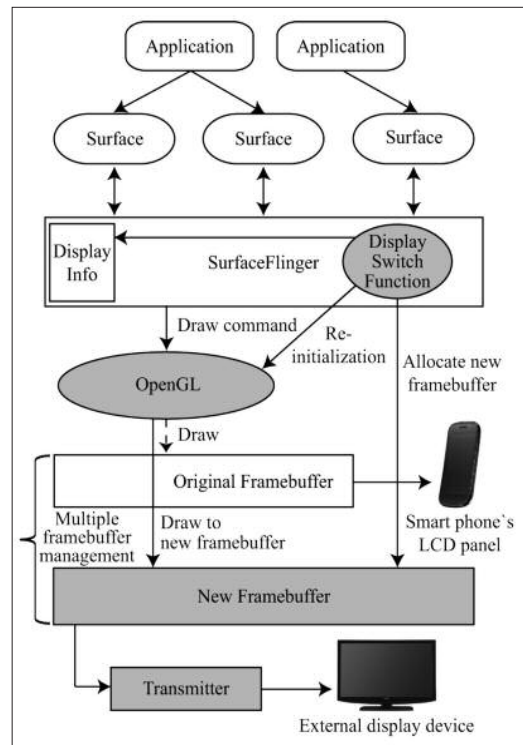
III. 적응형 디스플레이 전환 방법

사용자 주변에는 다양한 해상도를 가진 디스플레이 장치들이 사용되고 있다. 앞서 언급한 물리적 연결 기반의 전환 방법이나 응용 기반의 전환 방법들은 다양한 디스플레이 장치들이 가진 해상도를 최적으로 활용할 수 없었다. 따라서 스마트폰 중심의 N-Screen 기술의 핵심 요소 기술로써 디스플레이 전환 시 사용자 주변의 다양한 디스플레이 장치들을 최적 활용할 수 있는 적응형 디스플레이 전환 방법(Adaptive Display Switching Method)이 제안되었다[1].

〈그림 2〉는 현재 가장 널리 사용되고 있는 안드로이드 모바일 플랫폼의 디스플레이 구조를 나타



〈그림 2〉 기존 안드로이드 모바일 플랫폼의 디스플레이 구조



〈그림 3〉 적응형 디스플레이 전환 방법이 적용된 안드로이드 모바일 플랫폼의 디스플레이 구조

낸다. 안드로이드 모바일 응용은 화면을 그릴 수 있는 공간인 “Surface”들을 가지게 되는데, “SurfaceFlinger”라는 모듈에서 이 Surface들을 병합하여 화면을 구성하고, OpenGL을 호출하여 프레임버퍼에 그리면, 최종적으로 프레임버퍼의 내용이 스마트폰의 화면에 표시된다. 이러한 일련의 과정을 위해, 안드로이드 모바일 플랫폼은 해상도를 포함한 디스플레이 정보를 부팅 시에 SurfaceFlinger에 저장하고, 해상도에 따라 프레임버퍼를 할당한다.

하지만 이러한 디스플레이 정보들은 부팅 시에 한번 초기화되면 변경할 수 없으며, 할당된 프레임버퍼의 크기 또한 변경할 수 없다. 따라서 높은 해상도를 가진 새로운 외부 디스플레이 장치로 스마트폰 화면을 전환하더라도 기존 안드로이드 모바일 플랫폼이 가진 디스플레이 구조에서는 외부 디스플레이 장치가 가진 해상도를 최적으로 활용할 수 없다.

이러한 구조적 한계를 극복하기 위해 안드로이드

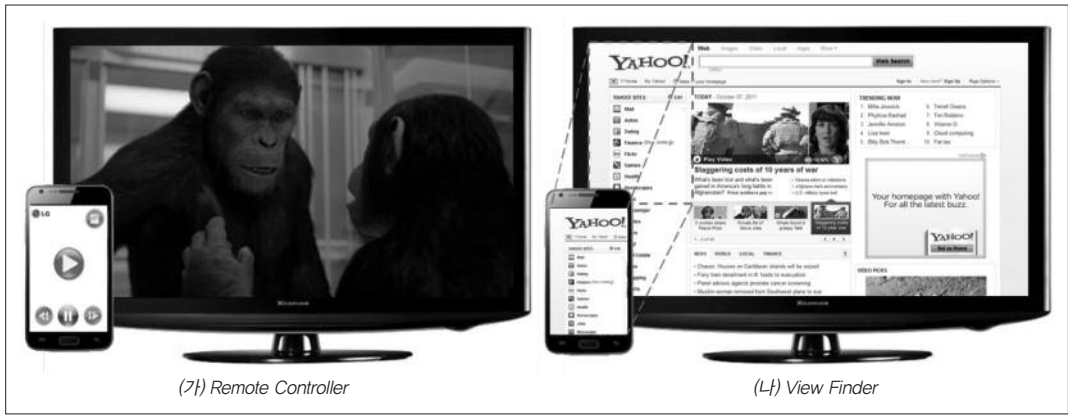
모바일 플랫폼 구조를 <그림 3>과 같이 적응형 디스플레이 전환이 가능하도록 수정하였다. 수정된 안드로이드 모바일 플랫폼에서는 부팅 시가 아니라도 디스플레이 정보와 프레임버퍼의 동적 재구성이 가능하다. 적응형 디스플레이 전환 방법에서는 사용자 주변에서 새로운 디스플레이 장치가 검색되면¹⁾ 자동으로 또는 사용자의 선택에 의해 디스플레이 전환을 시작한다. 이때 “DisplaySwitchFunction”은 외부 디스플레이 장치의 프로파일 정보에 따라 SurfaceFlinger의 디스플레이 정보를 업데이트하고, 해상도에 맞는 새로운 프레임버퍼를 할당하여²⁾ SurfaceFlinger가 새로운 해상도에 맞게 화면을 구성하여 알맞은 공간에 그릴 수 있도록 한다. 최종적으로 새로 할당된 프레임버퍼에 그려진 화면은 “Transmitter”에 의해 연속적으로 수집되어 전송되는 과정을 거쳐 외부 디스플레이 장치에 표시된다.

<그림 3>과 같이 적응형 디스플레이 전환 방법을 적용한 모바일 플랫폼에서는 자연스럽게 한 개 이



<그림 4> 디스플레이 전환 방법의 결과 비교

- 1) 적응형 디스플레이 전환 방법을 안드로이드 모바일 플랫폼에 적용한 데모에서는 Universal Plug and Play(UPnP)[10] 기술을 활용하여 사용자 주변의 디스플레이 장치를 검색한다.
- 2) 적응형 디스플레이 전환 방법에서는 메모리 맵이 고정되어 있는 시스템 메모리 영역이 아닌 사용자 메모리 영역을 활용하여 새 프레임 버퍼를 할당한다.



〈그림 5〉 적응형 디스플레이 전환 방법의 응용

상의 프레임버퍼를 가질 수 있다. 이러한 특징을 활용하여 개발자는 외부 디스플레이 장치의 효율적인 제어를 돕고 각 기기의 활용도를 높일 수 있는 다양한 응용들을 만들 수 있다. 예를 들어 미디어 플레이어의 경우, 평상시에는 스마트폰을 통해 동영상 재생하다가 외부 디스플레이 장치로 화면이 전환되면, 스마트폰의 내부 디스플레이 장치를 미디어 플레이어의 제어를 위한 Remote Controller로 활용할 수 있다.

IV. 안드로이드 모바일 플랫폼 기반의 N-Screen 기술 데모

스마트폰 중심의 N-Screen 기술 데모를 보이기 위해 적응형 디스플레이 전환 방법을 안드로이드 모바일 플랫폼 기반에 구현하였으며, 이를 Google의 Nexus S[3]에 실제 적용하였다. 이 기기는 Android 2.3.4 platform(Gingerbread) 기반이며, Linux kernel V2.6.35, Samsung Cortex A8

1GHz CPU, 그리고 512MB의 RAM으로 구성되어 있다.

먼저 기존의 디스플레이 전환 방법과 적응형 디스플레이 전환 방법 간의 차이점을 효과적으로 보이기 위해, 〈그림 4〉와 같이 스마트폰에서 HDTV로 화면을 전환하였다. 〈그림 4-(가)〉에서 모바일 디스플레이 장치는 전체 이미지의 일부분을 표시하고 있지만, 〈그림 4-(나)〉에서는 HDTV의 해상도에 맞게 화면이 재구성되어 전체 이미지가 HDTV 화면에 맞게 표시되었다. 반면에 물리적 연결 기반 디스플레이 전환 방법인 HDMI를 통해 스마트폰과 연결된 HDTV는 〈그림 4-(다)〉와 같이 스마트폰의 화면을 그대로 확대한 이미지만을 표시하였다. 또한 기존 응용 기반 전환 방법이나 DLNA를 이용한 방법과는 달리 적응형 디스플레이 전환 방법에서는 동영상 플레이어뿐만 아니라 웹 브라우저나 이미지 뷰어와 같은 일반적인 모바일 응용에서도 최적화된 형태로 화면이 표시된다.

〈그림 5〉는 적응형 디스플레이 전환 방법을 이용하여 스마트폰의 화면을 외부 디스플레이 장치로

전환한 이후, 스마트폰의 내부 디스플레이 장치를 모바일 응용에 따라 다양한 용도로 활용한 예를 보인다. <그림 5-(가)>는 외부 디스플레이 장치를 통해 동영상이나 이미지를 감상하면서 스마트폰의 내부 디스플레이 장치를 Remote Controller로 활용한 예로써, 미디어 플레이어뿐만 아니라 게임과 같은 응용에서도 매우 적합한 활용 예이다. <그림 5-(나)>는 외부 디스플레이 장치에 보여지는 화면 중 일부를 스마트폰의 내부 디스플레이 장치로 보면서 화면을 스크롤하거나 터치하여 모바일 응용을 제어할 수 있는 View Finder로 활용한 예로써, 넓은 화면 중 일부를 세밀하게 조작해야 하는 웹 브라우저나 맵 뷰어와 같은 응용에서 유용하게 사용할 수 있는 활용 예이다.

V. 결론

본 고에서는 스마트폰이 제공하는 다양한 서비스

를 사용자 주변에 있는 외부 디스플레이 장치를 활용해 보다 편리하게 제공하도록 하는 새로운 개념의 N-Screen 기술을 소개하였다. 또한, 이러한 스마트폰 중심의 N-Screen 기술의 핵심인 적응형 디스플레이 전환 방법(Adaptive Display Switching Method)에 대해 설명하고, 이를 안드로이드 모바일 플랫폼에 실제 적용한 데모를 보였다.

스마트폰 중심의 N-Screen 기술이 상용화 단계에 이르기 위해서는 아직도 많은 문제들이 선결되어야 한다. 특히 스마트폰과 같은 모바일 기기들이 가진 현재의 하드웨어 성능은 스마트폰 중심의 N-Screen 기술을 뒷받침할만한 수준에 이르지 못했다. 높은 해상도를 가진 외부 디스플레이 장치들의 화면에 표시될 이미지를 생산하면서도 이를 실시간으로 전송할만한 능력이 스마트폰에 요구되기 때문이다. 하지만 최근의 모바일 기기의 하드웨어 성능과 무선 네트워크 기술의 놀라운 발전 속도로 미루어 보았을 때, 이러한 하드웨어 성능상의 문제들은 가까운 미래에 자연스럽게 해결될 것으로 기대된다.

참고문헌

- [1] 권오철, 김주성, 이창건, 류혜정, "Modification of Android Platform for Seamless Convergence of Smart Phones and External Input/Output Devices", Android Competition in ESWEEK 2011, 2011년 10월
- [2] 윤장우, 전승현, 이호연, 허경우, 이현우, 류원, "N-스크린 서비스 분류와 표준화 기술", 정보과학회논문지 제29권 제7호 통권 제266호 pp.23-31 ISSN 1738-5385, 2011년 7월
- [3] Google, Nexus S, <http://www.google.com/nexus>
- [4] WirelessHD, WirelessHD, <http://www.wirelesshd.org>
- [5] Intel, Intel Wireless Display (WiDi), <http://www.intel.com>
- [6] Microsoft, Windows Remote Desktop Protocol, [http://msdn.microsoft.com/enus/library/windows/desktop/aa383015\(v=vs.85\).aspx](http://msdn.microsoft.com/enus/library/windows/desktop/aa383015(v=vs.85).aspx)
- [7] REALVNC, Virtual Network Computing (VNC), <http://www.realvnc.com>
- [8] DLNA, Digital Living Network Alliance (DLNA), <http://www.dlna.org>
- [9] Motorola, Motorola Atrix, <http://www.motorola.com/Consumers/US-EN/Consumer-Product-and-Services/Mobile-Phones/Motorola-ATRIX-US-EN>
- [10] UPnP Forum, Universal Plug and Play (UPnP), <http://www.upnp.org>

필자소개



권오철

- 2009년 2월 : 인하대학교 정보통신공학부 학사
- 2009년 3월 ~ 현재 : 서울대학교 컴퓨터공학부 석박통합과정
- 주관심분야 : *Mobile Platform, Adaptive UI Switching*



김주성

- 2010년 8월 : 서울대학교 컴퓨터공학부 학사
- 2011년 3월 ~ 현재 : 서울대학교 컴퓨터공학부 석사과정
- 주관심분야 : *Mobile Platform, Adaptive UI Switching*



이창건

- 1998년 8월 : 서울대학교 컴퓨터공학부 박사
- 1998년 3월 ~ 2000년 2월 : LG전자 선임연구원
- 2000년 3월 ~ 2002년 7월 : University of Illinois, Postdoctoral Research Associate
- 2002년 8월 ~ 2006년 8월 : Ohio State University, Assistant Professor
- 2006년 9월 ~ 현재 : 서울대학교 컴퓨터공학부 부교수
- 주관심분야 : *Real-time System, Cyber Physical System, Mobile Platform*