VNC 기반의 안드로이드 스마트 기기 화면 공유 시스템

박종은* · 이유동* · 이홍창** · 이명준***

VNC-Based Screen Sharing System for Android Smart Devices

Jong-Eun Park* · You-Dong Lee* · Hong-Chang Lee** · Myung-Joon Lee***

이 논문은 2011년도 울산대학교 연구비에 의하여 연구되었음

요 약

스마트폰과 스마트패드는 편리성과 휴대성을 비롯한 많은 장점들과 우수한 성능, 풍부한 어플리케이션을 바탕으로 정보를 효과적으로 관리할 수 있다. 이러한 스마트 기기의 활용 빈도가 늘어나면서 사용자들은 스마트 기기를 이용하여 다양한 정보를 관리하고 있다. 최근 회의, 세미나 중에 스마트 기기에 저장된 정보를 활용하고 시각적으로 공유하는 사례가 빈번히 발생하고 있지만 작은 크기의 디스플레이로 인하여 효과적으로 정보를 교환하기에는 근본적인 어려움이 있다. 본 논문에서는 다수의 안드로이드 스마트 기기의 화면을 하나의 대형 디스플레이 장치를 통하여 효과적으로 공유하는 기법을 제안하고 이를 바탕으로 스마트 기기 화면 공유 시스템을 개발하였다. 개발된 시스템은 여러 스마트 기기 화면을 통합하여 PC에 연결된 대형 디스플레이 장치에서 보여준다. 또한, 효과적인 공유를 위하여 대형 디스플레이 장치에서 출력되는 스마트 기기 화면의 개수와 그 화면 크기를 스마트 기기와 PC에서 직접 제어하는 기능을 제공한다.

ABSTRACT

Smartphones and smartpads have a lot of advantages such as high convenience and portability. Also, they can effectively manage enormous amounts of information based on their high performance and plentiful applications. As such devices have been frequently used, many users manage various types of information using the devices. Recently, during conferences or seminars, smart device users often try to utilize stored resources on their devices and present them visually. Unfortunately, since smart devices have small displays, there is an essential difficulty in visual sharing of information. In this paper, we propose a method which integrates screens of several android smart devices and supports sharing of the integrated screen through a large display, presenting a screen sharing system for android smart devices. The developed system integrates display screens of several smart devices into a screen and shows the integrated screen through a large display connected to a desktop computer. In addition, to support the effective sharing of screens, the system provides functions for adjusting the number and the size of screens on a large display. The functions are controlled on a smart device and/or a desktop computer.

키워드

VNC, Droid VNC Server, 화면 공유, 대형 디스플레이 장치, 안드로이드

Key word

VNC, Droid VNC Server, Screen Sharing, Large display devices, Android

* 준회원 : 울산대학교 컴퓨터정보통신공학부
** 정회원 : 울산대학교 컴퓨터정보통신공학부
*** 정회원 : 울산대학교 전기공학부 교수(교신저자, mjlee@ulsan.ac.kr)

접수일자 : 2012. 02. 02
심사완료일자 : 2012. 02. 06

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.3.522

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ⅰ. 서 론

회의, 학술 대회, 강의 등 한정된 장소에서 여러 사람들이 시각적 정보를 공유해야 하는 상황에서 대형 디스플레이 장치는 가장 흔하게 쓰이는 장치이다. 대부분 PC와 연동하여 PC에 저장된 정보를 보여주는데 이용되며, 사람들이 대형 디스플레이에서 보이는 화면을 보면서 서로의 의견을 교환하고 의사를 쉽게 결정할 수 있도록 활용되고 있다[1,2,3].

최근 다방면에서 활용되는 스마트 기기들은 편리성과 휴대성을 비롯한 많은 장점들과 우수한 성능, 다양한 기능을 제공하여 활용가치가 매우 높다. 그리고 이동 환경에 적합한 작은 크기로 인하여 언제, 어디서든 휴대하여 다양한 상황에서 적절하게 사용된다. 이러한 이유로 보급과 활용 빈도가 증가되어 사람들이 방대한 정보를 손쉽게 수집하고 수집된 정보를 효과적으로 관리할 수 있도록 한다. 최근 회의, 세미나에서 스마트 기기에 저장된 정보를 활용하고 시각적으로 공유하는 사례가 빈번히 발생하고 있다. 하지만 스마트 기기의 작은 디스플레이 크기로 인하여 효과적으로 정보를 공유하기에는 불편함이 있다. 따라서 스마트 기기의 시각적 정보를 여러 사람들과 효과적으로 공유하는 시스템이 지원된다면 매우 편리할 것이다

본 논문에서는 다수의 안드로이드 스마트 기기의 화면을 대형 디스플레이 장치에서 효과적으로 공유할 수 있는 기법을 제안하고 이를 바탕으로 개발된 스마트 기기 화면 공유 시스템에 대해 기술한다.

개발된 시스템은 원격지 PC를 제어하는데 널리 사용되는 오픈소스 소프트웨어인 VNC(Virtual Network Computing)[4]를 기반으로 구현되었으며, PC에서 동작하는 서버와 스마트기기 기반의 클라이언트로 구성된다. 서버는 스마트 기기의 화면을 PC에 연결된 대형 디스플레이서 보여줄 수 있는 환경을 지원하며, 서버에 접속한 스마트 기기의 화면의 개수와 크기를 손쉽게 제어할 수 있는 기능을 제공하여 다수의 스마트 기기 화면을 효과적으로 관리할 수 있다. 클라이언트는 스마트 기기의 화면을 서버로 전송하고, 서버를 통해 디스플레이 장치에 출력된 스마트 기기의 화면 크기를 직접 제어하는 기능을 제공하며 관리자에 의해서만 실행된다. 또한, 관리자 권한 양도, 화면의 특정 부분 강조, 스마트 기기에 부착된 카메라의 화면 전송 등 다양한 기능을 제공한다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 VNC와 스마트 기기 화면 공유 시스템에 대한 배경지식을 다루고, 3장에서는 다수의 스마트 기기 화면 공유를 위한 기법을 제시하며, 4장에서는 제시된 기법을 바탕으로 개발한 시스템에 대해 설명하고, 5장에서는 개발된 시스템과 타 시스템을 비교하여 설명한다. 마지막으로 6장에서는 결론을 다룬다.

Ⅱ. 배경 지식

2.1. 스마트 기기의 VNC 시스템

VNC는 원격지 PC를 제어할 수 있는 그래픽 기반의 시스템이다. 1999년 AT&T에서 오픈 소스로 제공되어 다양한 플랫폼에서 동작하는 VNC 기반 시스템이 등장하였다[5]. 주로 사용자가 위치한 로컬 환경에서 원격지에 있는 PC를 제어하기 위한 목적으로 개발되었으며 대표적인 시스템은 Tiger VNC[6], Ultra VNC[7], Real VNC[8] 등이 있다. VNC는 서버와 클라이언트로 구성되며, 서버는 화면 정보를 클라이언트에게 전달하는 역할을 한다. 클라이언트는 VNC 서버로부터 전달받은 화면 정보를 사용자가 볼 수 있도록 표현하고, 사용자의 입력을 서버로 전송하는 역할을 한다. 이러한 서버와 클라이언트의 통신은 RFB(Remote Framebuffer)[9] 프로토콜을 기반으로 수행된다.

최근에는 스마트 기기에서 동작하는 VNC 기반의 여러 시스템이 개발되었다. 대부분 스마트 기기에서 원격지 PC를 제어할 수 있는 형태로 제공 되며, 대표적인 어플리케이션은 Pocketcloud[10], Remote VNC[11] 등이 있다. 이러한 어플리케이션은 VNC 클라이언트로 개발되어 기존의 VNC 시스템과 유사한 형태로 동작한다. 즉, 스마트 기기에서 VNC 서버가 설치된 원격지 PC를 제어할 수 있는 기능을 제공한다. 이러한 이유로 이동환경에서 편리하게 원격지 PC의 작업 진행 정도를 확인하거나 새로운 작업 명령을 지시할 수 있다. 그리고 이와 같은 기능을 효과적으로 지원하기 위하여 PC의 높은 해상도를 스마트 기기 화면에서 충분히 표현하기 위한 연구도 활발히 진행되고 있다.[12,13]

2.2. 스마트 기기 화면 공유 시스템

최근에는 스마트 기기의 정보를 시각적으로 많은 사람과 공유하는 상황이 많아짐으로써 스마트 기기를 PC에서 제어할 수 있는 시스템이 등장하고 있다. 이러한 시스템은 작은 디스플레이의 스마트 기기 화면을 PC와 연결되는 대형 디스플레이 장치에서 실시간으로 보여줌으로써, 스마트 기기의 시각적 정보를 효과적으로 공유할 수 있다. 특히, 스마트 기기의 디스플레이 고장, 터치 고장 등 치명적인 문제가 발생할 때에도 유용하게 사용된다. 이러한 기능을 제공하는 대표적인 시스템은 VNC 기반의 Droid VNC Server[14]와 USB 케이블 이용하는 ASM(Android Screen Monitor)[15] 등이 있다.

Droid VNC Server는 안드로이드 플랫폼의 스마트 기기에서 동작하는 VNC 기반의 어플리케이션으로 RFB 프로토콜을 지원한다. 구조는 스마트 기기의 화면을 전송하는 Droid VNC Server와 Tiger VNC를 확장한 PC 환경의 클라이언트로 구성된다. Droid VNC Server는 스마트 기기에서 백그라운드 서비스로 동작하며 프레임버퍼의 화면 정보를 PC의 클라이언트로 전송한다. 클라이언트에서는 전송된 스마트 기기의 화면을 실시간으로 확인하며 스마트 기기를 제어하는 기능을 제공한다. 그리고 PC의 키보드를 이용하여 스마트 기기의 메뉴, 전화 등의 버튼을 입력하는 기능을 지원한다.

ASM은 ADB(Android Debug Bridge)[16]의 도구로 안드로이드 기반의 스마트 기기 화면을 PC에서 제어하기 위한 목적으로 개발 되었다. ADB는 안드로이드 기반 스마트 기기와 통신하며 스마트 기기의 디버깅을 위한 클라이언트-서버 프로그램이다. 통신 방식은 PC와 스마트 기기 간 USB 연결을 통해 수행되며 PC의 ADB 클라이언트를 통해 스마트 기기를 제어한다. 이러한 ASM은 화면 공유를 위하여 스마트 기기에 별도의 어플리케이션을 설치하지 않기 때문에 스마트 기기 기반의 어플리케이션 개발 발표 등의 상황에서 유용하게 사용되고 있다.

Ⅲ. VNC 기반의 스마트 기기 화면 공유 기법

3.1. 기법 및 시스템 구성

기존의 화면 공유를 위한 방법은 스마트 기기가 서버가 되어 화면 정보를 전송하고, PC의 클라이언트에서 전송된 화면을 출력하는 형태를 보인다. 이러한 형태에서 여러 스마트 기기 화면을 공유하기 위하여 Droid VNC Server는 스마트 기기 수만큼 PC 클라이언트가 수작업으로 실행되어야 하며, ASM 역시 스마트 기기 수만큼 USB 케이블을 연결해야 한다. 이는 대단히 비효율적인 작업이며, 다수의 스마트 기기를 관리하기에도 불편하다. 따라서 다수의 스마트 기기 화면을 하나의 대형 디스플레이에서 효과적으로 보여줄 수 있는 방법과 이를 지원하는 시스템이 필요하다.

본 연구에서는 다수의 안드로이드 스마트 기기 화면을 효과적으로 대형 디스플레이 장치에서 보여주기 위하여 그림 1과 같은 형태로 동작하는 기법을 제안한다.

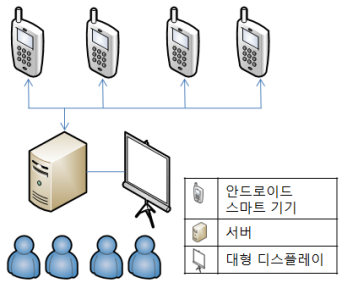

그림 1. 다수의 스마트 기기 화면 공유 방식
Fig. 1 Method of Screen Sharing for Smart devices

그림 1에서는 하나의 서버와 여러 대의 스마트 기기가 서로 통신하며, 스마트 기기는 화면을 서버로 전송한다. 서버는 수신된 화면을 연결된 대형 디스플레이 장치에서 여러 가지 형태로 보여준다. 이러한 형태는 다수의 스마트 기기가 하나의 서버를 통해 관리될 수 있기 때문에 기존의 화면 공유 방법의 문제점을 해결할 수 있다. 그림 2는 제안한 형태에서 동작하는 시스템의 내부 구조를 보여준다.

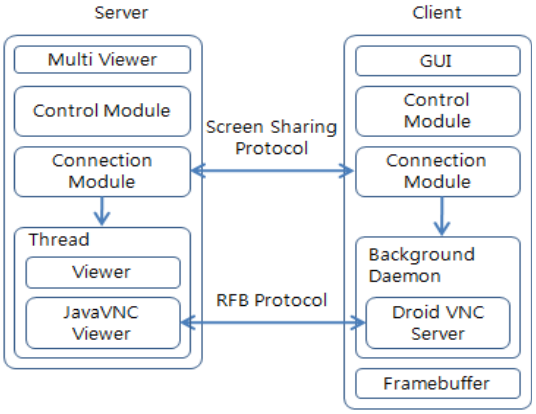

그림 2. 시스템 구조
Fig. 2 Structure of the Proposed System

그림2에서 서버와 클라이언트는 화면 전송과 출력을 위하여 VNC를 이용한다. 서버는 VNC 클라이언트 개발을 지원하는 Java VNC Viewer를 통해 화면 정보를 수신하여 출력한다. 클라이언트는 서버에 화면을 전송하기 위하여 Droid VNC Server를 기반으로 한다.

서버는 연결된 여러 스마트 기기의 화면을 효과적으로 보여주는 멀티 뷰어(Multi Viewer)를 가진다. 멀티 뷰어는 서버에 접속한 다수의 스마트 기기 화면을 통합하여 보여주며, 상황에 맞는 크기 및 배치 조절 기능을 지원한다. 연결 모듈(Connection Module)은 클라이언트와 통신하며 클라이언트의 요청을 처리한다.

그리고 클라이언트가 접속하면 스레드를 생성하여 Java VNC Viewer를 실행한다. 실행된 Java VNC Viewer는 스마트 기기 화면을 전송받아 뷰어(Viewer)에 출력하고, 뷰어는 멀티 뷰어에 의해 관리되며 사용자에게 스마트 기기 화면을 보여준다.

클라이언트의 GUI는 사용자와 상호작용하며, 사용자로부터 전달되는 요청을 제어 모듈(Control Module)로 전달한다. 제어 모듈은 GUI를 전환하거나 서버로 전송되는 요청을 클라이언트의 연결 모듈로 전송한다. 그리고 서버와 연결이 완료되면 화면 전송을 위하여 스마트 기기 백그라운드에서 Droid VNC Server를 동작시킨다. Droid VNC Server는 프레임버퍼에 접근하여 화면 정보를 실시간으로 서버로 전송한다. 이러한 클라이언트와 서버 사이에서 화면을 전송하고 서비스를 종료하기 위한 과정은 다음과 같다.

(1) 사용자가 클라이언트의 GUI를 통해 서버와 연결을 요청
(2) 클라이언트의 연결 모듈에서 서버와 연결을 하고 백그라운드에서 Droid VNC Server를 동작시킴
(3) 클라이언트의 접속을 확인한 서버는 스레드를 생성하여 Java VNC Viewer를 통해 클라이언트에서 실행된 Droid VNC Server와 연결
(4) 연결된 VNC 간에 FRB 프로토콜을 통해 화면 정보를 서버에서 수신하고 수신된 화면을 멀티 뷰어로 출력
(5) 사용자가 클라이언트의 GUI를 통해 서버와 연결 종료를 요청
(6) 클라이언트의 연결 모듈에서 종료 요청을 서버로 전송하고 백그라운드에서 실행된 Droid VNC Server를 종료 시킴
(7) 서버에서 종료 요청을 확인하고, 해당 클라이언트의 스레드를 종료

3.2. 효과적인 화면 공유 서비스

서버는 여러 클라이언트의 접속을 처리하고, 클라이언트로부터 전송되는 화면을 상황에 따라 배치, 크기 등을 조절하여 화면 공유 상황을 효과적으로 지원하는 것이 바람직하다. 그리고 클라이언트의 접근을 제한하거나, 강제로 종료시키는 등의 하여 보안상의 문제를 다룰 수 있어야 한다. 본 논문에서는 효과적인 화면 공유를 위한 서비스를 표1과 같이 정의하였다.

표 1. 화면 공유 서비스 목록
Table. 1 List of Screen Sharing Services

범주	기능
서버	화면 제어 서비스 - 멀티 뷰어 배치 조절 - 스마트 기기 화면 크기 조절 클라이언트 관리 서비스 - 클라이언트 접속 종료 - 관리자 권한 양도 - 관리자 권한 제거
클라이언트	통신 서비스 - 서버 연결 - 접속 종료 - 클라이언트 목록 요청 화면 관련 서비스 - 화면 강조 - 카메라 뷰어

서버에서는 화면 제어 서비스와 클라이언트 관리 서비스를 제공하여 로컬 환경에서 손쉽게 다수의 클라이언트와 그 화면을 제어할 수 있다. 클라이언트는 효과적인 화면 공유를 위한 화면 관련 서비스와 통신 서비스를 제공한다. 통신 서비스는 클라이언트에서도 서버의 서비스가 실행되도록 지원한다. 이를 위하여 클라이언트에서 서버의 서비스를 요청하는 프로토콜 메시지를 표2와 같이 정의하였다.

표 2. 화면 공유 프로토콜 메시지
Table. 2 Protocol Message for Screen Sharing

요청 정보	설명
Connect	서버 접속
Disconnect	서버 접속 종료
Ask_Wiser	특정 화면 확대
Ask_Closer	특정 화면 축소
Change_Manager	관리자 권한 양도
Self_Mode	관리자 기능 중지
Ask_Client_List	모든 클라이언트 정보 요청
Comparing	멀티 뷰어의 모든 화면을 동일한 크기로 배치
Arrange	멀티 뷰어의 모둔 화면을 기본 형태로 배치
Purge	특정 클라이언트 강퇴

표2에서 정의된 메시지를 기반으로 통신하는 시스템은 여러 클라이언트의 중복 요청으로 인하여 화면과 클라이언트의 관리가 어려워지는 문제가 발생할 수 있다. 이러한 문제를 해결하기 위하여 클라이언트의 역할을 설정하고, 그 역할에 따라 수행되는 기능에 제한을 두었다. 역할은 관리자와 일반 사용자로 구분되며, 관리자 역할의 클라이언트에서 서버로 프로토콜 메시지를 전송하여 서버의 서비스가 수행되도록 하였다.

Ⅳ. VNC 기반의 스마트 기기 화면 공유 시스템의 개발

4.1. 화면 공유 서버

서버는 클라이언트의 연결 및 다양한 요청을 처리하기 위하여 Java 소켓을 기반으로 구현하였다. 화면 수신은 Java VNC Viewer를 이용하여 연결된 대형 디스플레이 장치에 보여준다. Java VNC Viewer는 RFB 프로토콜을 교환하는 FtbProto 클래스, 수신된 화면을 보여주는 VncViewer 클래스 등의 화면 공유에 있어서 필수적인 클래스를 제공한다. VncViewer 클래스는 수신된 화면을 프레임(JFrame)에 출력하도록 구현되어 있다.

이는 하나의 화면을 출력하기에는 적합하지만 다수의 화면을 보여주기에는 프레임의 수가 많아짐으로 효과적인 화면 공유를 지원하지 못한다. 이를 위하여 VncViewer 클래스를 프레임 안에서 프레임의 역할을 지원하는 인터널프레임(JInternalFrame)에 화면이 출력이 되도록 변경 하였다. 그림 3은 변경된 소스 코드의 일부를 보여준다.

```
JinternalFrame jf = ne JinternalFrame(userName, ture, ture,
ture, ture);
...
 fr.getContenPane().add(vc);
 fr.setBounds(x,y,width,height);   fr.setVisible(true);
 //멀티뷰어에 인터널프레임을 삽입
 canvasPanel.add(fr);
```

그림 3. 다중 화면을 지원하는 변경된 Vnc Viewer 클래스
Fig. 3 Modified Vnc Viewer Class Supporting Multi Screen

변경한 VncViewer 클래스를 이용하며, 손쉬운 화면 제어를 지원하는 서비스와 클라이언트 관리 서비스를 위하여 표 3과 같은 역할을 수행하는 클래스를 구현하였다.

표 3. 개발된 서버 클래스
Table. 3 Developed Classes in the Server

클래스	역할
Server	서버 자원을 초기화하고 클라이언트의 접속을 처리한다.
ControlFrame	화면 제어 서비스, 클라이언트 관리 서비스를 제공하는 인터페이스와 그 기능을 수행
ClientManager	접속한 클라이언트를 관리하며, 관리자와 일반 사용자를 구분한다.
ClientSocket	Java VNC Viewer를 동작시키며, 클라이언트와의 연결 상태와 화면 공유 서비스의 프로토콜을 처리한다.
Protocol	프로토콜의 정보를 제공한다.

표 3에서 Server 클래스는 서버 자원을 초기화 하는 과정에서 서버의 인터페이스를 구성하며 멀티 뷰어를 생성하여 클라이언트를 기다린다. Protocol 클래스와 ClientSocket 클래스는 통신 모듈을 구성하며, 클라이언트와 연결되어 프로토콜 메시지를 처리한다. Server 클래스와 통신 모듈을 이용하여 클라이언트의 화면을 수신하는 과정은 다음과 같다.

(1) Server 클래스가 동작하여 클라이언트를 기다림
(2) 클라이언트가 접속하면 Server 클래스에서 클라이언트를 처리하기 위한 ClientSocket을 스레드로 실행
(3) ClientSocket 클래스에서 클라이언트의 프로토콜 메시지를 기다림
(4) 클라이언트가 “Connect” 프로토콜 메시지를 전송
(5) ClientSocket 클래스에서 Protocol 클래스를 참조하여 수신된 메시지를 확인
(6) ClientSocket 클래스에서 화면 수신을 위하여 Java Vnc Viewer의 VncViewer 클래스의 객체를 생성
(7) 생성된 VncViewer 객체는 화면 수신을 위한 작업을 수행하고 클라이언트의 화면을 Server 클래스의 멀티 뷰어로 보낸다.

그림 4는 이러한 동작과정을 구현한 소스코드의 일부를 보여준다.

```
public class ClientSocket extends Thread{
...
  if(message.equals(Protocol.CONNECT)){
    //ip와 사용자 정보를 저장
    String ip = socket.getInetAddress().getHostN
    String userName = br.readLine();
    //클라이언트 관리와 로그, 제어를 위한 설정
    clientManager.add(socket, userName);
    controlFrame.addUser(userName);
    controlFrame.printLog("[ "+userName+" ] 님이
    접속했습니다.");
   //접속 완료메시지 전송
    pw.println(Protocol.CONNECT_SUCCESS)
    //Java VNC Viewer를 실행
    vnc = new VncViewer(vncCanvas, ip);
    vnc.init(screen_width, scrren_hight, pX, pY);
...
```

그림 4. Java VNC Viewer 실행 처리 구현
Fig. 4 Implementation of Handling Java VNC Viewer execution on the Server

그림3에서 VncViewer 클래스는 화면 수신을 처리하며 멀티 뷰어에 출력될 위치와 크기를 설정한다. 이는 ControlFrame에서 멀티 뷰어에 출력되는 모든 뷰어의 출력 지점과 크기를 저장하고, 비교를 통해 수행될 수 있도록 하였다.

멀티 뷰어에서 수행되는 화면 제어 서비스는 서버에서 직접 수행되거나 클라이언트로부터 전송된 프로토콜 메시지에 의해서 실행된다. 이를 위하여 Control Frame 클래스에 기능 수행에 필요한 메소드를 표 4와 같이 정의하고, 사용자의 요청이나 프로토콜 메시지에 따라 처리되도록 하였다.

표 4. 화면 제어 서비스를 지원하는 메소드
Table. 4 Methods Supporting for Screen Controlling Services

메소드	역할
void wiser()	특정 클라이언트 화면 확대
void closer()	특정 클라이언트 화면 축소
void comparing()	2개 이상의 화면을 동일한 크기로 멀티뷰어에 배치

표 4의 메소드는 그림 5와 같이 구현하였으며, Client Manager 클래스의 객체에서 해당 클라이언트를 불러와서 인터널프레임의 크기를 변경하기 위하여 정의한 adjustScrrenSize() 메소드를 통하여 화면 제어 서비스를 실행한다.

```
//확대 화면 구현
public void wiser(){
  //해당 클라이언트를 불러옴
  clientScoket = clientManager.getClient(index);
  //가로 길이는 멀티 뷰어/3, 높이는 멀티 뷰어 높이
  //출력 지점은 X,Y 좌표가 0으로 설정
  clientScoket.getVnc().adjustScrrenSize(
frame_width/3, frame_height, 0, 0);
}
//확대 축소 구현
public void close(){
  //해당 클라이언트를 불러옴
  clientScoket = clientManager.getClient(index);
  //출력 지점과 가로, 세로 크기가 원래 크기로 설정
  clientScoket.getVnc().adjustScrrenSize(
screen_width, screen_height, pX, pY);
}
```

그림 5. 화면 제어 서비스의 구현
Fig. 5 Implementation of Screen Controlling Services

4.2. 화면 공유 클라이언트

클라이언트에서 화면 정보를 서버로 전달하기 위하여 Droid VNC Server를 이용하였다. Droid VNC Server는 스마트 기기의 프레임버퍼에 접근하여 화면 정보를 전송하는데, 프레임버퍼는 안드로이드의 보안상의 문제로 접근이 제한되어 있다.

따라서 Droid VNC Server를 실행하기 위해서는 프레임버퍼에 접근하는 방법이 우선적으로 시행되어야 하므로 스마트 기기 루팅(Rooting)[17]이 필요하다. 루팅은 리눅스를 기반으로 하는 안드로이드 플랫폼에서 디바이스, 파일 시스템, 프레임버퍼 등에 접근하기 위하여 관리자 권한을 얻는 행위를 말한다. Droid VNC Server는 화면 전송을 위하여 초기화 작업을 수행하는 MainApplication 클래스, 백그라운드에서 화면을 실시간으로 전송하는 ServerManager 클래스 등을 제공한다.

MainApplication 클래스는 프레임버퍼의 화면 정보를 읽기 위한 유틸리티 파일들을 스마트 기기 파일 시스템에 복사한다. 유틸리티 파일들은 루팅된 스마트 기기에서 실행되는 다양한 리눅스 명령어들과 프레임버퍼에 접근하여 화면 정보를 전송하는 androidvncserver[18] 등이 있다. 이러한 유틸리티들은 ServerManager 클래스와 상호작용하여 화면 전송에 관련된 작업을 수행한다.

Droid VNC Server를 이용하여 화면을 전송하는 클라이언트는 서버와 통신하며 수행되는 여러 서비스를 제공한다. 이를 위하여 표 5와 같은 클래스를 정의하고, 이러한 서비스를 지원하는 GUI를 구현하였다.

표 5. 화면 공유 서비스를 위한 클래스
Table. 5 Classes for Screen Sharing Services

클래스	역할
CameraActivity	스마트 기기의 카메라를 동작시켜 카메라의 화면을 보여준다.
ControlActivity	멀티 뷰어를 제어하기 위한 GUI로 사용자로부터 액션을 입력받는다.
HighlighterActivity	사용자로부터 화면을 강조하기 위한 액션을 입력받고, 그에 관련된 서비스를 제공한다.
DialogActivity	대화 상자로 출력되어 상황에 맞는 GUI 클래스를 실행한다.
ClientSocket	서버와 연동되어 연결 상태 및 각종 요청을 처리한다.
Protocol	프로토콜의 정보를 제공한다.

HighterActivity에서는 사용자와 상호작용하며, 현재의 화면에서 다양한 효과를 부여하는 화면 강조 서비스를 제공한다. 화면 강조 서비스는 사용자의 터치를 이용하여 화면의 특정 부분에 색상을 넣거나, 도형을 그리도록 지원한다. 이러한 기능을 구현하기 위하여 현재의 화면 위에 투명한 화면이 배치되도록 하였다. 사용자는 투명한 화면을 통해 현재의 화면을 보고 강조기능은 그림 6과 같이 투명한 화면에서 수행된다.

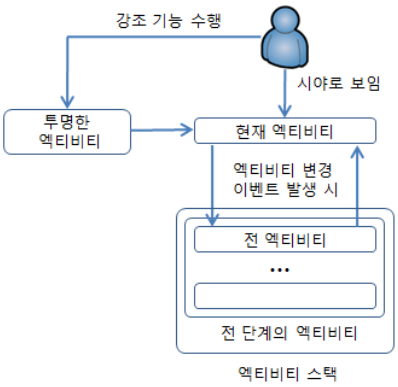

그림 6. 강조 기능 기법
Fig. 6 Technique for Highlight Function

V. 구현 결과 및 기능 비교

5.1. 구현 결과

개발된 시스템은 하나의 서버에서 수신된 다수의 스마트 기기 화면을 대형 디스플레이 장치에서 효과적으로 보여줌으로써 이를 통해 스마트 기기에 저장된 정보를 많은 사람들과 시각적으로 공유하도록 지원한다. 개발된 서버에서는 멀티 뷰어를 통해 다수의 스마트 기기 화면을 분할하여 보여주고, 로그와 제어 서비스를 통해 효과적으로 클라이언트를 관리할 수 있다. 그림 7은 개발된 서버의 모습을 보여준다.

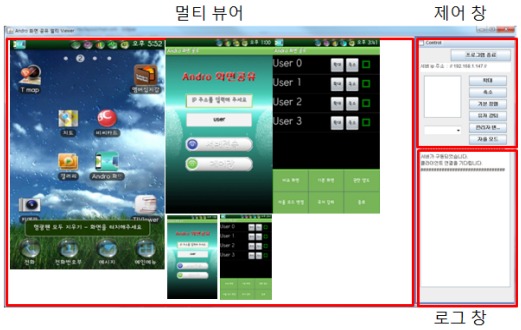

그림 7. 개발된 서버의 인터페이스
Fig. 7 Interface of the Developed Server

그림 6에서 로그 창은 클라이언트의 연결과 요청에 관련된 로그를 보여준다. 제어 창은 클라이언트 관리 서비스와 화면 제어 서비스의 손쉬운 실행을 지원하는 인터페이스로 화면 제어서비스의 요청을 멀티 뷰어로 전달한다. 화면 제어 서비스는 특정 클라이언트의 확대 및 축소, 2개 이상의 화면을 동일한 크기로 변경하는 비교 화면 기능을 지원한다. 멀티 뷰어는 제어 창에 의해 요청된 화면 제어 서비스를 즉각적으로 그림 8과 같이 반영하여 스마트 기기 화면의 배치와 크기를 조절한다.

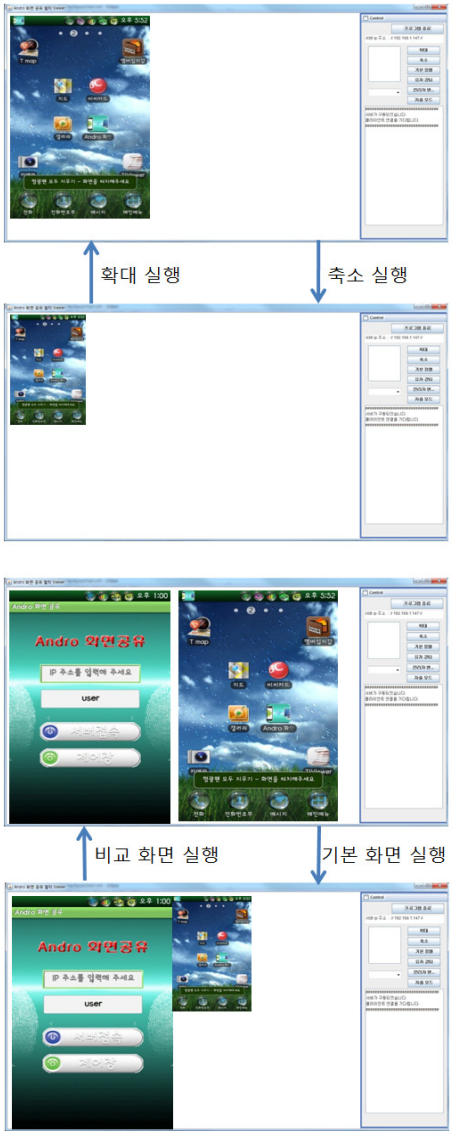

그림 8. 화면 제어 서비스의 화면
Fig. 8 Example of Screen Controlling Service

개발된 클라이언트는 스마트 기기의 백그라운드에서 실행되어 화면을 서버로 전송하며, 서버와 연동하여 화면 제어, 클라이언트 관리 서비스를 제공한다. 그리고 화면 관련 서비스를 지원하여 시각적 자료를 보다 효율적으로 공유할 수 있다. 그림 9는 구현된 클라이언트에서 수행되는 강조기능, 클라이언트 관리 및 제어 화면을 보여준다.

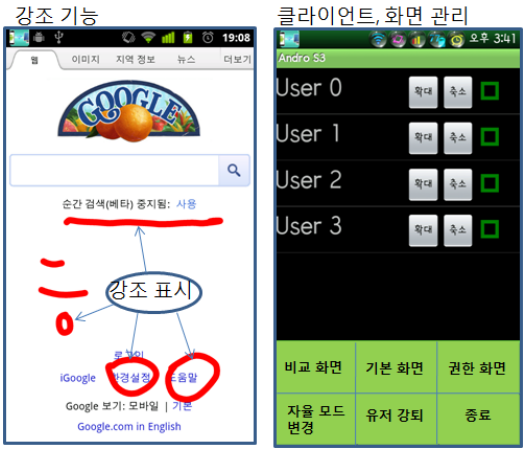

그림 9. 다양한 클라이언트 기능
Fig. 9 Various Client Functions

그림 10은 개발된 시스템을 활용하여 스마트 기기의 자료를 시각적으로 공유하는 모습을 보여준다.

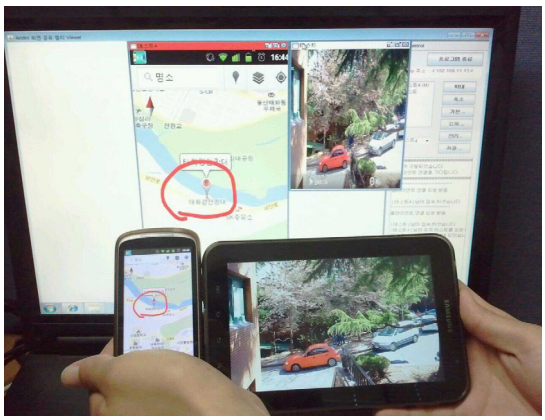

그림 10. 개발된 시스템의 활용
Fig. 10 Using the Developed System

5.2. 타 시스템과의 기능 비교

Droid VNC Server는 VNC를 기반으로 하는 대표적인 스마트 기기 화면 공유 시스템으로 RFB 프로토콜을 통해 효과적인 스마트 기기 제어 서비스를 제공한다. 하지만 다수의 화면을 공유하기 위해서는 PC에서 구동되는 클라이언트가 스마트 기기 수만큼 수작업으로 실행되어야 하므로 관리 측면에서 비효율적이며, 다중 사용자 화면을 지원하지 않는다.

ASM은 ADB와 연동되어 동작하는 스마트 기기 화면 공유 시스템으로 루팅을 하지 않아도 실행 가능한 장점이 있다. 그러나 USB 케이블을 이용해야지만 통신이 가능한 단점과 함께 Droid VNC Server와 마찬가지로 다중 사용자 화면을 보여주지 못하기 때문에 다양한 상황에서 활용되기에는 많은 어려움이 따른다.

본 연구에서 개발된 시스템은 다중 사용자 화면과 함께 화면 관리 서비스를 제공한다. 따라서 작은 디스플레이를 가지는 스마트 기기의 불편함을 해소하고, 여러 사람들 간에 다수의 스마트 기기의 화면을 효과적으로 공유할 수 있다. 그리고 안드로이드 정책상 프레임버퍼에 접근하기 위하여 루팅이 필요하지만 향후 정책이 변경된다면 루팅 과정이 생략되어 더 널리 활용될 것이다. 표 6은 개발된 시스템과 여러 스마트 기기 화면 공유 시스템의 기능 비교를 보여준다.

표 6. 여러 스마트 기기 화면 공유 시스템의 특징
Table. 6 Features of each Screen Sharing System for Smart Devices

	Droid VNC Server	ASM	개발된 시스템
화면 공유	O	O	O
통신 방식	소켓 통신	USB 연결	소켓 통신
루팅 여부	O	X	O
다중 사용자 화면	X	X	O
화면 관리 서비스	X	X	O
화면 강조 서비스	X	X	O

Ⅵ. 결론

본 논문에서는 다수의 안드로이드 스마트 기기의 화면을 대형 디스플레이 장치에서 효과적으로 공유할 수 있는 기법을 제안하고 이를 바탕으로 개발한 스마트 기기 화면 공유 시스템에 대해 기술하였다. 개발된 시스템은 VNC를 기반으로 구현되었으며 PC에서 동작하는 서버와 스마트기기 기반의 클라이언트로 구성된다.

서버는 소켓통신으로 클라이언트와 연결되며, 클라이언트에서 전송되는 화면을 연결된 대형 디스플레이 장치로 출력한다. 그리고 클라이언트의 요청에 따라 화면 관리를 수행한다. 클라이언트는 화면 정보를 서버로 전송하고, 서버에 연결된 디스플레이 장치에 출력되는 화면을 손쉽게 제어할 수 있는 인터페이스와 기능을 제공한다. 특히, 화면 강조 기능을 지원하여 스마트 기기의 시각적인 정보를 효과적으로 공유할 수 있다.

개발된 시스템은 다중 사용자 화면과 화면 관리 서비스를 지원하여 스마트 기기의 시각적 정보를 여러 사람들과 효과적으로 공유하도록 하여 스마트 기기에 저장된 정보를 활용하는 회의, 세미나 등의 다양한 상황에서 널리 활용될 수 있다.

감사의 글

본 연구는 2011년도 울산대학교의 지원에 의하여 이루어진 연구로서, 관계부처에 감사 드립니다.

참고문헌

[1] E.S.Abdulmotaleb. R.Abdur, A.Souhail, “PECOLE: P2P multimedia collaborative environment”, Multimedia tools and applications v.39 no.3, pp.353-377, 2008

[2] 이태호, 박지혜, 이기훈, 이명준, “CoDisplay : 대형 디스플레이 장치를 활용하기 위한 VNC 기반회의 도구”, 한국해양정보통신학회논문지 제 13권 제 8호, pp.1666-1672, 2009

[3] 김남곤, 김종원, “Visual Sharing: 다자간 원격 협업 환경에서의 View 공유 기술”, 한국HCI학회 2008년도 학술대회 1부 2008 Feb. 13, pp.643-647, 2008

[4] T.Richardson, Q.Stafford-Fraser, K.R.Wood and A.Hoper, ″Virtual Network Computing″, IEEE Internet Computing, Volume 2. Number 1, January/February, pp33-38, 1998

[5] “http://www.corp.att.com/attlabs/”, AT&T Labs

[6] “http://www.tightvnc.org/”, TigerVNC

[7] “http://www.uvnc.com/”, UltraVNC

[8] “http://www.realvnc.com/”, RealVNC

[9] T.Richardson, ″The RFB Protocol″, 2006

[10] “http://www.wyse.com/products/software/pocketcloud/android/”, PocketCloud

[11] “http://www.toremote.com/”, Remote VNC

[12] 김태훈, 최종찬, 이정준, ″3D 디자인을 위한 안드로이드 TurboVNC 뷰어″, 한국정보처리학회 2011년도 제35회 춘계학술발표대회 2011 Apr. 30, pp.1135-1138, 2011

[13] A.Skurski B.Swiercz ″VNC-based remote control for Symbian OS smartphones″, Mixed Design of Integrated Circuits&Systems, 2009. MIXDES ‘09. MIXDES-16th International Conference 2009 June, pp.171-174 , 2009

[14] “https://github.com/oNaiPs/droid-VNC-server/”, Droid VNC Server

[15] “http://code.google.com/p/android-screen-monitor/”, Android Screen Monitor

[16] “http://developer.android.com/guide/developing/tools/adb.html/”, Android Debug Bridge

[17] “http://en.wikipedia.org/wiki/Rooting_(Android_OS)”, Rooting(Android OS)

[18] “http://code.google.com/p/android-vnc-server/”, android-vnc-server

저자소개

박종은(Jong-Eun Park)

2011년 울산대학교 컴퓨터정보통신공학부 졸업(학사)
현재 울산대학교 정보통신공학 석사과정

※관심분야 : 소셜네트워크 서비스, 클라우드 시스템
※Email : cjswowhddms@nate.com

이유동(You-Dong Lee)

2012년 울산대학교 컴퓨터정보통신공학부 졸업(학사)

※관심분야 : VNC 시스템, 클라우드 시스템
※Email : udong85@mail.ulsan.ac.kr

이홍창(Hong-Chang Lee)

2006년 울산대학교 컴퓨터정보통신공학부 졸업(학사)
2008년 울산대학교 컴퓨터정보통신공학부 졸업(석사)
2010년 울산대학교 컴퓨터정보통신공학부 박사 수료
※관심분야 : 클라우드 시스템, 웹 서비스, 소셜네트워크 서비스
※Email : myhyunii@mail.ulsan.ac.kr

이명준(Myung-joon Lee)

1980년 서울대학교 수학과 졸업 (학사)
1982년 한국과학기술원 전산학과 졸업(석사)
1991년 한국과학기술원 전산학과 졸업(박사)
1993 ~ 1994년 미국 버지니아대학 전산학과 교환교수
2005 ~ 2006년 미국 캘리포니아 주립대학 교환교수
1982 ~ 현재 울산대학 컴퓨터정보통신공학부/전기공학부 교수
※관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래밍 시스템, 소셜네트워크 서비스
※Email : mjlee@ulsan.ac.kr