

논문 2010-3-5

서비스 로봇 제어를 위한 안드로이드 기반의 유비쿼터스 인터페이스

Android Based Ubiquitous Interface for Controlling Service Robots

전영훈*, 안현식**

Yongxun Quan, Hyunsik Ahn

요 약 본 논문에서는 안드로이드 기반의 서비스 로봇 제어를 위한 유비쿼터스 인터페이스를 제안한다. 로봇의 호스트 컴퓨터인 서버에서 전방의 영상, 지도 정보와 로봇의 현재 위치를 표현한 그래픽, 및 로봇의 모션에 대한 그래픽 영상을 안드로이드 클라이언트에 전달하고, 안드로이드는 그 결과를 화면상에 표현한다. 안드로이드는 로봇을 원격제어하거나 목표위치 지정 및 로봇의 동작을 제어하기 위한 정보를 버튼으로 표현하고 그 정보를 서버에 전달하여 안드로이드 기반의 유비쿼터스 인터페이스가 가능하도록 한다. 로봇의 이동과 동작을 제한된 크기의 스마트폰 화면에서 제어하기 위해 원격 이동 모드, 자동이동 모드 및 원격조작 모드를 구현한다. 실험에서는 모토로이 안드로이드 폰을 이용한 인터페이스를 구현하여 실제 서비스 로봇 제어에 적용한 결과를 통해 제안한 인터페이스가 로봇 제어를 위한 유비쿼터스 인터페이스로서 응용될 수 있음을 보여준다.

Abstract In this paper, an Android based ubiquitous interface for controlling service robots is presented. The robot server captures the images for the front view of the robot, makes a map of the environment and its position, produces a graphic image of its pose, and then transmits them to the Android client. The Android client displays them in the LCD panel and transfers control information obtained from touched buttons to the server. In the interface environment, we implement remote moving mode, autonomous moving mode, and remote operation mode for being used for versatile operability to the robot with limited screen of the smart phone. Experimental results show the implementation of the proposed interface in Android installed on Motoroi to control a service robot, and demonstrate its feasibility.

Key Words : Android, Smart Phone, Ubiquitous Control, Service Robot, Tele-operation

I. 서 론

서비스 로봇을 시간과 장소의 제약 없이 제어하기 위한 유비쿼터스 환경에 대한 관심이 지속되고 있다^[1,2]. 최근 스마트폰의 등장으로 다양한 유비쿼터스 애플리케이션에 대한 기대가 높아지고 있는데, 앱스토어 등장과 함께 다양한 애플리케이션 개발이 되고 있다^[3,4]. 스마트폰은 서비스 로봇을 제어하기 위한 유비쿼터스 인터페이스

장치로서도 충분히 활용될 수 있는 가능성을 가지고 있다. 즉 실내에서 동작하는 서비스 로봇을 제어 하기 위해 기존의 컴퓨터에서 나아가 스마트폰을 이용하면 언제 어디서나 로봇에 대한 제어가 가능한 유비쿼터스 제어가 가능하다고 할 수 있다. 기존의 로봇에 대한 유비쿼터스 인터페이스 환경에 대한 연구에서는 휴대폰의 Wipi 등을 이용하여 이동로봇의 제어를 위한 인터페이스의 적용의 예가 있다^[5]. 그러나 제한된 컴퓨팅 환경과 고가의 네트워크 사용료로 인해 실제로 거의 활용되고 있지 않다. 이에 비해 스마트폰은 Wi-Fi를 이용할 경우 사용료가 없으

*준회원, 동명대학교 로봇시스템공학과

**정회원, 동명대학교 로봇시스템공학과(교신저자)

접수일자 2010.06.15 수정일자 2010.06.25

며, 보다 개선된 컴퓨팅 환경을 제공하므로 스마트폰이 서비스 로봇용 인터페이스로 상용하기에 적절하다고 할 수 있다.

본 논문에서는 안드로이드(Android) 기반의 서비스 로봇 제어를 위한 유비쿼터스 인터페이스를 제안한다. 로봇의 호스트 컴퓨터인 서버에서 전방의 영상과 지도 정보와 로봇의 현재 위치를 표현한 그래픽과 로봇의 모션에 대한 로봇의 동작을 표현한 결과를 안드로이드 클라이언트에 전달하고, 안드로이드는 그 결과를 화면상에 표현한다. 안드로이드는 로봇을 원격제어하거나 목표위치 지정 및 로봇의 동작을 제어하기 위한 정보를 버튼으로 서버에게 전달하여 안드로이드 기반의 유비쿼터스 사용자 인터페이스가 가능하도록 한다.

본 논문에서는 먼저 서비스 로봇 제어하기 위한 인터페이스의 시스템 구성을 설명하고, 서비스 로봇 제어용으로 활용하고자 할 경우의 안드로이드 기반의 스마트폰의 특성과 안드로이드 기반의 프로그래밍 방법론과 제한된 스마트폰의 스크린에서 다양하게 로봇을 제어하기 위한 제어 모드에 대해 설명한다. 이어서 실제 서비스 로봇의 제어를 위한 스마트폰 인터페이스를 구현한 실험 결과를 보인 후 결론을 맺는다.

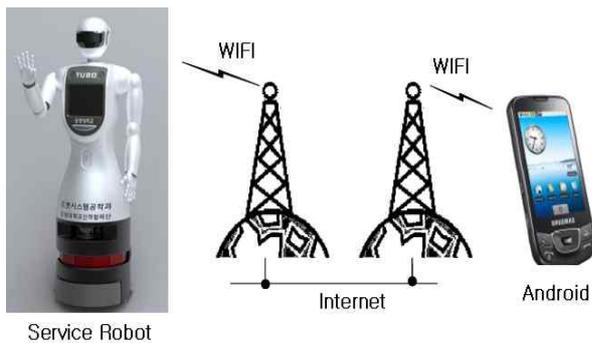


그림 1. 안드로이드 기반의 서비스 로봇 제어를 위한 유비쿼터스 인터페이스의 구조
 Fig. 1. The schematic of the Android based ubiquitous interface for controlling service robots

II. 유비쿼터스 인터페이스의 시스템 구성

본 장에서는 서비스 로봇을 제어하기 위한 유비쿼터스 인터페이스의 기본 아이디어와 시스템 구성을 설명한

다. 그림 1은 본 논문에서 제안하는 스마트폰 기반의 로봇 제어 인터페이스의 개념을 보여주고 있다. 언제 어디서나 스마트폰으로 서비스 로봇에게 접근하여 자유롭게 제어할 수 있는 유비쿼터스 인터페이스가 가능하며 Wi-Fi기반으로 통신을 함으로써 무료이면서도 양질의 고속 서비스가 가능하다.

제안하는 인터페이스를 환경을 구성하고 네트워크 및 제어를 구현하기 위해 컴퓨팅 조건을 적절히 배분한다. 스마트폰의 안드로이드에서는 제한된 컴퓨팅 환경을 고려하여 인터페이스에 치중하고 서버 상에서는 컴퓨팅 기능을 주로 담당하도록 설계한다. 즉 로봇이 적절한 형태로 가공한 인터페이스 데이터를 전달하면 안드로이드는 그 결과를 화면에 표현하거나 버튼을 이용한 이벤트를 발생시키는 역할을 한다. 또한 스마트폰의 LCD 화면은 데스크탑 모니터에 비해 작은 공간이지만 로봇의 움직임을 제어하는데 있어서 실수를 배제할 수 있을 만큼의 충분한 조작의 용이성이 필요하다. 따라서 사용자가 입력시에 화면상의 임의의 위치를 터치하기 보다는 분명한 구분이 가능한 버튼이 보다 적절할 수 있으므로 터치스크린을 이용하되 버튼을 구현하여 입력 화면 인터페이스 장치로 사용한다.

일반적으로 서비스 로봇은 가정에서 인간이 필요로 하는 일들을 서비스 할 수 있는 기능을 가지고 있어야 한다. 본 논문에서 테스트 베드로 활용하고자 하는 서비스 로봇 TUBO(Tongmyong University roBOt)는 서비스 제공을 위해 필수적인 기능을 가지고 있는데, 자기위치 검출과 지도 작성(SLAM), 로봇 전방 영상 표시, 로봇 이동 경로 계획, 자율적 이동과 위기 회피, 원격조작 기능 등이다. 서비스 로봇의 호스트 컴퓨터는 SBC(Single Board Computer)를 사용하였으며, 이동 플랫폼은 다사 로봇의 TETRA-DS를 사용한다. 센서로는 자기위치 검출을 위해 laser range scanner인 Sick사의 LMS-200을 장착하였으며 영상입력용 카메라로 Bumblebee 스테레오 카메라를 사용하였다. 실내에서 이동하며 서비스를 제공하기 위해서는 로봇의 자기 위치검출(localization)이 필요한데 먼저 실내 공간에 대한 지도가 주어지며, LMS-200에 의해 로봇을 중심으로 한 전방향의 거리 데이터를 검출하고 이를 지도와 정합하여 자기위치를 검출한다. 안드로이드 개발을 위한 플랫폼으로는 Motorola의 Motoroi를 사용하며 유비쿼터스 인터페이스 개발 환경은 Android2.0을 사용한다.

III. 안드로이드 기반의 로봇 제어 인터페이스

1. 안드로이드의 특성

안드로이드는 운영체제와 미들웨어 그리고 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택이다. 안드로이드의 주요 특징은 다음과 같다^[6,7].

- 애플리케이션 프레임워크 컴포넌트에 대한 재사용 및 치환이 가능함.
- Dalvik 가상머신 모바일 디바이스를 위해 최적화됨
- 통합 브라우저 오픈 소스 WebKit 엔진이 제공됨
- 최적화된 2D 그래픽 라이브러리와 OpenGL ES 1.0 기반의 3D 그래픽 라이브러리를 제공
- SQLite로 통해 정형화된 데이터 저장 공간 제공
- 미디어를 위해 오디오, 비디오 그리고 정지 이미지 포맷들을 지원
- 블루투스, EDGE, 3G와 WiFi 가능
- 디바이스 에뮬레이터, 디버깅 툴, 메모리 및 성능 프로파일링 그리고 Eclipse IDE를 위한 플러그인을 제공

그림 2는 안드로이드 운영체제의 주요 구성 요소를 보여주고 있는데, 애플리케이션, 애플리케이션 프레임워크, 라이브러리, 리눅스 커널 등의 계층적 구조로 이루어져 있으며 애플리케이션은 Java를 사용하여 작성한다^[7].



그림 2. 안드로이드의 아키텍처
Fig. 2. Android architecture

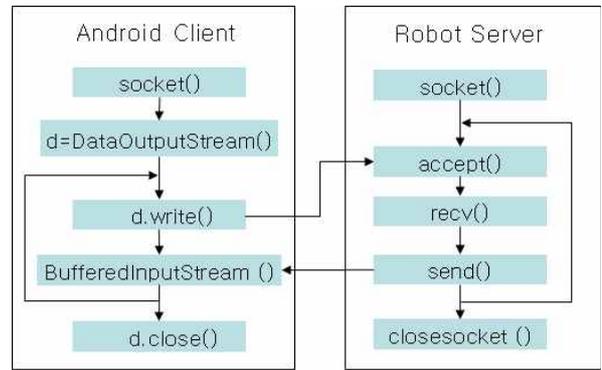


그림 3. 인터페이스 네트워킹
Fig. 3. Interface networking

2. 안드로이드 기반 인터페이스 프로그래밍

본 절에서는 로봇 제어 인터페이스를 위한 네트워킹과 안드로이드 기반 프로그래밍 방법에 대하여 설명한다. 먼저 네트워킹을 살펴보면, 안드로이드를 기반으로 하여 로봇 제어 인터페이스를 구현하기 위한 로봇 서버와 안드로이드 클라이언트 사이의 네트워크의 기능을 표현하면 그림 3과 같다. 로봇 제어 인터페이스는 서버에서 만들어진 영상 또는 그래픽을 보여주는 기능과 클라이언트에서 로봇 제어를 위한 버튼 이벤트를 처리하는 것으로 구분할 수 있다. 영상 송신을 위해서는 서버에 Socket()을 설정하고 대기하고 있으면, 클라이언트 상에서 DataOutputStream()을 이용하여 신호를 보내면 서버는 준비된 영상이나 그래픽 파일을 영상포맷을 만들어 send()로 전송하며 클라이언트는 BufferedInputStream()으로 받는다. 버튼 이벤트 신호는 클라이언트에서 d.write()로 보내면 서버에 전송하면 서버 상에서 recv()로 받는다.

다음으로 로봇 제어 인터페이스를 위한 안드로이드 상에서 프로그래밍 방법에 대하여 설명한다. 먼저 영상을 서버로부터 전달 받고 이를 표시하기 위해 그림 4와 같이 안드로이드 상에서 onCreate() 함수에서 imageSwitcher를 생성하여 서버로부터 받은 영상을 연속적으로 표시할 수 있도록 한다. 이와는 별도로 Thread()를 이용한 while 루프를 사용하며, 그 내부에 Socket()을 생성한 후, BufferedInputStream()으로 영상을 연속으로 받아들이고 Bitmap으로 처리한 후 이를 setImageDrawable()로 연속적으로 표시한다.

```

imageSwitcher.setFactory(new ImageSwitcher.ViewFactory(){
    public View makeView(){
        ImageView iv = new ImageView(RemoteControl.this);
        iv.setBackgroundColor(0xFF000000);
        iv.setLayoutParams(new
ImageSwitcher.LayoutParams(450,300));
        return iv;
    }
});
    
```

그림 4. 영상 수신 및 표시를 위한 함수
 Fig. 4. The function for receiving and displaying images

본 논문에서는 클라이언트의 입장에서 영상 신호의 수신과 버튼을 이용한 제어 신호의 송신을 별도의 네트워크 포트를 이용하여 처리함으로써, 버튼 이벤트가 보다 고속으로 서버에게 전달되도록 한다. 버튼 이벤트의 처리는 그림 5와 같이 myClickHandler() 함수를 사용하는데 영상 송수신과는 별도의 Socket()을 생성하고 버튼에 해당하는 변수를 파라미터로 하여 d.write()함수로 전달한다. 안드로이드 상에서의 버튼의 구성은 setContentView()를 호출하여 이루어 지는데 버튼의 형태는 layout 디렉토리의 main.xml에서 설정한다.

```

public void myClickHandler(View target) {
    Socket s;
    DataOutputStream dos;
    s = new Socket();
    dos = new DataOutputStream(s.getOutputStream());
    switch (target.getId()) {
        case R.id.Button08:
            command = 8;
            setTitle("8");
            break;
    }
    dos.write(pack);
    dos.close();
    s.close();
}
    
```

그림 5. 버튼 이벤트 전송 함수
 Fig. 5. The function for transmitting button events

3. 서비스 로봇의 제어 모드

본 절에서는 제한된 스마트폰의 스크린에서 로봇을 보다 효율적으로 제어하기 위한 원격이동 모드, 자동이동 모드, 원격조작 모드의 기능을 설명한다. 안드로이드 스마트폰은 하단에 3가지 모드의 버튼이 있어서 사용자

의 필요에 따라 선택할 수 있다.

1) 원격이동 모드

원격이동 모드는 로봇의 이동을 스마트폰의 스크린에 나타나는 버튼으로 로봇의 이동을 제어하는 모드이다. 사용자가 로봇의 전방의 환경을 인지할 수 있도록 로봇 전방의 카메라 영상을 입력하여 클라이언트로 전달하여 그 결과를 스마트폰 스크린에 표시한다. 또한 명령 버튼을 화면에 표시하여 버튼이 눌러졌을 때 그 정보를 서버에게 전달하여 요구하는 명령을 실행하도록 한다. 버튼은 로봇의 이동에 필요한 기능으로서 전진, 후진, 좌회전, 우회전 및 정지로 구분하고, 클라이언트에서 이동 버튼이 눌러질 때 마다 로봇 서버는 그 방향으로 속도를 증가시키고, 반대 방향의 버튼이 눌러지면 속도를 감소시키도록 하여 로봇의 이동에 대한 제어가 인간의 인지적 판단에 따르도록 설정한다. 따라서 원격이동 모드에서는 전방의 상황을 영상으로 관찰하면서 버튼으로 이동을 제어할 수 있다.

2) 자동이동 모드

자동이동 모드에서는 로봇의 현재 위치와 향후 이동할 목표 위치를 그래픽으로 표현하는 모드이다. 로봇 서버에 로봇의 활동 영역에 대한 그리드 지도(grid map)를 제공한다. 로봇에게는 전방에 레이저 거리 스캐너가 부착되어 있으며 전방에 대한 거리 정보를 이용하여 로봇의 자기 위치를 검출(localization) 알고리즘을 이용하여 구한 현재의 로봇 위치 (x, y, θ)를 모니터 상에 표현한다. 자기위치 검출을 위해 그리드 지도와 로봇의 거리 검출 스캐너를 이용한 거리 데이터를 정합(matching)하여야 하는데 본 논문에서는 일반적으로 많이 활용되고 있는 몬테카를로 기법(Monte Carlo method)을 이용한다. 로봇의 자세가 얻어지면 이를 지도상에 그래픽으로 표현한 후 그 영상을 클라이언트로 전달하여 표현하게 한다. 사용자는 지도상에서 현재 로봇의 위치를 확인할 수 있고 버튼을 이용하여 로봇의 이동 목표 위치를 지도상에서 표현하고 그 정보를 서버에게도 전달한다. 버튼은 전방, 후방, 좌, 우를 이용하여 이동한 후 선택 버튼으로 목표 위치가 결정되도록 하여 임의의 위치를 목표로 설정할 수 있도록 한다. 로봇은 내부적으로 목표 위치로 이동하기 위한 최적 경로를 추정하며 지도 상에 로봇의 위치를 갱신시키면서 이동한다.

3) 원격조작 모드

원격 조작모드는 로봇의 팔, 손, 머리 부분을 원격으로 조작하여 원하는 작업을 하기위한 모드이다. 원격조작이 되도록 하기 위해 먼저 로봇의 역기구학(inverse kinematics)을 풀어서 로봇의 팔과 손이 (x, y, z)에 의해 위치 제어가 가능하도록 한다. 안드로이드를 이용한 조작을 위하여 로봇 호스트 컴퓨터에서는 로봇에게 로봇의 머리에 부착된 카메라로부터 전방의 영상을 입력하여 전송하고, 로봇의 조작과 관련된 모터의 기구학적 외형을 평면과 측면 자세를 그래픽으로 표현하여 클라이언트에 전달한다. 안드로이드 인터페이스는 로봇의 원격 조작과 관련된 양팔과 양손 및 머리 부분에 대한 제어 버튼을 설치한다. 로봇팔과 손에 대한 제어를 위해 좌우 X, Y, Z 축과 손의 개폐 및 머리의 상하 좌우 움직임에 해당하는 버튼을 배치한다. 로봇의 조작은 다수의 입력 기능이 필요한데, 제한된 스크린의 공간에서 입력 기능을 최대한 제공하기 위해 원격조작 모드에서는 기능 버튼과 이동 버튼을 구분하여 2개의 이벤트에 의해 동작이 이루어 지도록 한다. 즉 로봇의 오른팔을 X축 전방으로 이동시키고자 할 경우 먼저 'RX' 버튼을 누르고 '+'버튼을 눌러서 이동시킨다.

IV. 실험 및 결과

본 논문에서는 제안한 안드로이드 기반 로봇 제어 인터페이스를 실험하기 위하여 서비스 로봇을 테스트 베드로 하고 실제 안드로이드 스마트폰을 이용하여 구현하였다. 그림 6은 본 실험에서는 사용된 서비스 로봇 TUBO이며 이를 제어하는 로봇 서버는 SBC이고 개발 환경은 VC++을 이용하였다. 스마트폰은 Motorola의 Motoroi를 사용하였으며 인터페이스 개발 환경은 Android2.0을 사용하였다. 그림 5는 제안한 안드로이드 기반 로봇 제어 인터페이스를 실제 스마트폰으로 구현하고 제어실험을 한 결과이다. 그림 7(a)는 원격이동 모드로서 사용자가 직접이동을 지시하는 경우에 사용할 수 있으며, 좌편의 로봇 전방의 영상을 관찰하면서 버튼을 이용하여 로봇의 이동을 제어할 수 있다. 그림 7(b)는 자동이동 모드로서 지도 영상에 로봇의 위치를 적색으로 표현하고 있으며 버튼을 이용하여 적색의 원으로 로봇의 이동 목표를 지시하고 있다. 그림 7(c)는 원격조작 모드

로서 로봇의 팔과 머리 등을 제어할 수 있으며 좌편에는 현장의 영상과 함께 로봇의 자세를 그래픽으로 표현하여 사용자가 로봇 제어하는데 로봇의 자세를 추정할 수 있도록 하였다. 원격조작 모드에서는 한 동작을 위해 동작 버튼과 방향버튼(+, -)을 동시에 활용하도록 하여 제한된 버튼으로 충분한 기능을 발휘할 수 있도록 하였다.

본 실험에서의 결과로 무선 네트워크의 QoS에 따라 달려있기는 하나 초당 5 frame 정도의 영상이나 그래픽의 통신이 가능하였으며 버튼을 통한 이벤트 처리는 50msec로 처리가 가능하였다.



그림 6. 실험을 위한 서비스 로봇 (TUBO)
Fig. 6. The service robot for the experimentation (TUBO)

V. 결론

본 논문에서는 서비스 로봇을 제어하기 위한 안드로이드 기반의 인터페이스 방법을 제안하고 실제로 구현하였다. 먼저 서비스 로봇과 안드로이드 스마트폰으로 구성된 유비쿼터스 제어 환경을 구성하고 안드로이드 클라이언트에 버튼과 영상으로 이루어진 인터페이스를 구성하였다. 로봇 서버에서 입력된 영상이나 지도 또는 로봇 자세에 대한 그래픽을 클라이언트에 전달하였으며, 버튼을 이용한 클라이언트의 제어 신호를 서버에 전달하여 사용자가 언제 어디서나 로봇의 제어가 가능하도록 하였다. 스마트폰의 제한된 조건 하에서 원격이동 모드와 자동이동 모드 및 원격조작 모드를 제어의 특징에 따라 구성하여 보다 용이하게 로봇에 대한 제어를 가능하게 하였다. 실험을 통해 제안한 인터페이스 환경이 사용자가 언제 어디서든지 로봇을 제어 할 수 있는 유비쿼터스 환

경이 구성되었다고 할 수 있다.

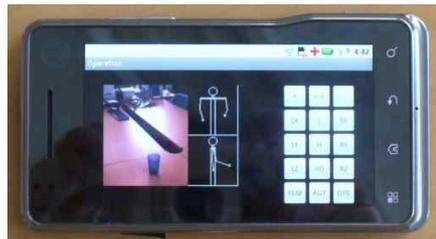
추후 연구 방향은 스마트폰이 제공하고 있는 다양한 컴퓨팅 환경을 충분히 이용하여 보다 인간 친화적이고 인지적으로 용이한 인터페이스를 구성하는 것이라 할 수 있다. 이를 위해 안드로이드에서 제공하는 3D 그래픽 등을 이용한 가상현실 또는 증강현실(augmented reality)을 이용한 방법도 고려되어야 하며 음성의 전달과 인식 기능으로 원격지에서 로봇과 통화하면서 제어하는 것도 보다 개선된 유비쿼터스 환경으로 필수적 기능이라고 할 수 있을 것이다.



(a)



(b)



(c)

그림 7. 안드로이드 스마트폰에서의 로봇 제어 인터페이스 모드 (a) 원격이동 모드, (b) 자동이동 모드 (c) 원격조작 모드

Fig. 7. Modes of Android interface for controlling the robot (a) Remote moving mode, (b) autonomous moving mode, (c) Remote operation mode

참 고 문 헌

- [1] H.J. Lee and C.S. Leem, "A study on value chain in a ubiquitous computing environment," Lecture Notes in Computer Science, vol. 3483, pp. 113-121, 2005.
- [2] 조영록, 오상록, "지능형 서비스 로봇과 URC (Ubiquitous Robotic Companion)," 한국통신학회, 제12권 제10호, pp. 13-21, 2004.
- [3] 오범교, 강태환, 안병구, "안드로이드 기반 모바일 Booky 콘텐츠," 한국인터넷방송통신학회논문지, 제10권, 제2호, pp.53-60, 2010.
- [4] 배성호, 김우생, "안드로이드 기반 모바일 정보공유시스템," 대한전자공학회 논문지-CI, 제46권 제2호, pp. 58-64, 2009.
- [5] 안현식, 송재성, "실내 환경에서의 이동로봇 제어를 위한 유비쿼터스 인터페이스 시스템," 제어·자동화·시스템공학논문지, 제12권, 제1호, pp. 66-71, Jan. 2006.
- [6] 김정훈, "구글의 안드로이드와 안드로이드마켓," 한국콘텐츠학회지, 제7권, 제2호, pp. 29-36, 2009.
- [7] <http://developer.android.com/index.html>

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2010-0013740)

저자 소개

전 영 훈(준회원)



- 2005년 연변대학 계산기과학과 학사
- 2010년~현재 동명대학교 로봇시스템 공학과 석사과정
- 2005년~2008년 연길비트정보유한회사 엔지니어
- 2008년~2009년 Capinfo Company Limited 엔지니어

<주관심분야 : 지능로봇, 로봇제어, 로봇비전>

안 현 식(정회원)



- 1986년 경북대 전자공학과 학사
- 1989년 동대학원 전자공학과 석사
- 1998년 동대학원 박사
- 1992년~1998년 포항산업과학연구원 선임연구원
- 2007년 미국 GeorgiaTech 방문교수
- 1998년~현재 동명대학교 로봇시스템 공학과 부교수

<주관심분야 : 지능로봇, 로봇비전, 인지기구조, HRI, 서비스로봇>