

인터넷을 이용한 이동로봇의 원격 운용 시스템

Teleoperation System of a Mobile Robot over the Internet

박 태 현, 강 근 택, 이 원 창

(Taehyun Park, Geuntaek Kang, and Wonchang Lee)

Abstract : This paper presents a teleoperation system that combines computer network and an autonomous mobile robot. We control remotely an autonomous mobile robot with vision over the Internet to guide it under unknown environments in the real time. The main feature of this system is that local operators need a web browser and a computer connected to the communication network and so they can command the robot in a remote location through the home page. The hardware architecture of this system consists of an autonomous mobile robot, workstation, and local computers. The software architecture of this system includes the client part for the user interface and robot control as well as the server part for communication between users and robot. The server and client systems are developed using Java language which is suitable to internet application and supports multi-platform. Furthermore, this system offers an image compression method using JPEG concept which reduces large time delay that occurs in network during image transmission.

Keywords : teleoperation, mobile robot, Internet, Java, JPEG

I. 서론

최근 인터넷의 급속한 성장은 우리가 정보를 수집하고 전송하는데 있어서 공간적 한계나 시간상 제약을 극복할 수 있는 통신 수단이 되고 있다. 또한 인터넷에 연결된 카메라, 망원경, 매니플레이터, 이동로봇 등과 같은 장치들이 최근에 속속 등장하고 있을 뿐만 아니라 인터넷에 접속되는 시스템의 종류가 점점 늘어날 전망이다. 이와 같이 인터넷을 이용한 원격운용 및 모니터링 시스템은 사용자에게 원격지의 정보를 획득, 감시, 특정임무 수행 등과 같은 특별한 기회를 제공함으로써 시간적으로나 공간적으로 많은 효율성을 기대할 수 있다. 원격운용 및 모니터링 시스템의 활용 분야는 오염지역 탐사, 핵 산업, 군수 산업 같이 인간이 작업할 수 없는 위험한 환경에서의 정보 획득 및 임무 수행이나, 수중 탐사, 우주탐사와 같이 미지의 환경에서의 정보수집 등 여러 분야에 적용될 수 있다. 한편 컴퓨터 네트워크를 통한 로봇의 원격운용 또한 현실에 적용할 수 있는 잠재적인 이용 가치가 있는 시스템으로서 중요한 연구 주제가 되었으며, 웹 브라우저를 통해 로봇 매니플레이터를 원격지에서 제어하고자 하는 여러 연구가 진행되었으며[1]-[3]. 이를 이동로봇에 적용하고자 하는 연구결과도 최근에 보고되고 있다[4]-[6].

인터넷 기반의 이동로봇 원격운용 시스템에서는 로봇 제어를 자연스럽게 적용할 수 있는 로봇 소프트웨어뿐만 아니라 사용자와 로봇 사이의 효과적인 데이터 전송방식을 개발하는 것이 중요하다고 볼 수 있으며, 이러한 시스템 개발에 있어 Java는 새로운 표준언어로서 부각되고 있다. 통신 수단으로서 인터넷을 사용하는 원격운용 시스템은 인터넷 지향

적 그리고 멀티 플랫폼을 지원하는 언어로서 소프트웨어가 구성되어야 한다. 이러한 필요성을 가장 잘 만족시키는 언어가 바로 Java이다. Java Applet으로 제공되어진 사용자 인터페이스는 인터넷으로 연결된 플랫폼에서 웹 브라우저를 통하여 실행되며, Java Application은 어떠한 수정이나 재 컴파일 없이 JVM(Java virtual machine)이 설치된 어떤 기계에서나 동작할 수 있는 플랫폼 독립성과 호환성을 지원한다. 또한 객체 지향적 언어로서 강력한 컴퓨터 네트워크 프로그램, 사용자를 위한 보안성 보장, 그래픽 환경 애플리케이션의 대화적인 성능 향상을 위한 다중 쓰레드 등을 지원하기 때문이다[7]-[9].

본 논문에서는 이동로봇을 원격지에서 운용함으로써 기존의 기능에서 탈피한 능동적인 운용을 하고자 하며, 인터넷을 이용함으로써 운용 공간의 한계를 극복하고자 한다. 또한 인터넷 사용자와 이동 로봇과의 양방향 통신을 통하여 로봇과 친밀할 수 있는 기회를 제공함과 동시에 원격지의 영상 정보를 실시간으로 공유하고자 하는 것이 연구의 목적이다. 본 연구에서는 로봇 시스템, 서버 시스템, 클라이언트 시스템으로 이루어진 원격운용 시스템 구축, 이미지 전송을 위한 압축, TCP/IP 소켓을 이용한 네트워크 프로그램, DC모터 제어알고리즘을 실제로 구현하여 원격지의 자율이동 로봇을 실시간으로 운용하고 그 가능성을 검증하고자 한다.

II. 원격운용 시스템

1. 원격운용 시스템의 개요

원격운용 시스템은 크게 서버와 클라이언트 개념의 컴퓨터 네트워크와 자율이동 로봇구조를 가지며 제안된 전체 시스템은 앞에서 기술한 바와 같이 서버 시스템, 클라이언트 시스템, 로봇 시스템으로 나눌 수 있으며 이것을 나타내면 그림 1과 같다. 원격운용 시스템에서 서버 시스템, 클라이언트 시스템은 네트워크 기능, 보안, 데이터 베이스, 이중 플랫폼간의 호환성에 있어 강력한 Java로 구현되어지며, 로봇 시

접수일자 : 2001. 5. 11., 수정완료 : 2001. 12. 5.

박태현 : (주)에이앤에스(hyena@ans.co.kr)

강근택 : 부경대학교 전자컴퓨터정보통신공학부(gtkang@pknu.ac.kr)

이원창 : 부경대학교 전자컴퓨터정보통신공학부(wlee@pknu.ac.kr)

※ 본 논문은 2000년도 부경대학교발전기금의 지원에 의하여 연구되었습니다.

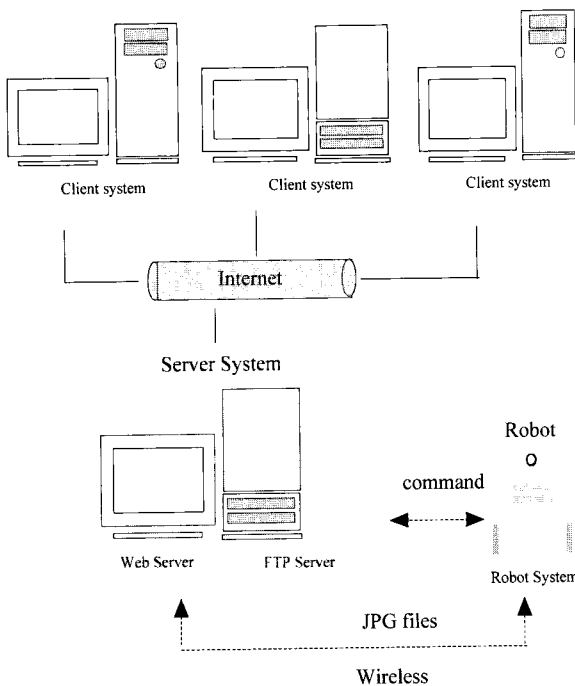


그림 1. 이동로봇 원격운용 시스템의 전체 구조.
Fig. 1. Overall structure of teleoperation system for a mobile robot.

시스템은 윈도우즈 환경에서 멀티쓰레드, 객체지향, GUI 기능이 뛰어난 윈도우 프로그래밍 언어인 MFC로 구현된다.

이러한 시스템의 주된 특징은 사용자가 자율이동 로봇을 운용하기 위해서 일반적인 목적의 컴퓨터와 웹 브라우저가 제공되어야 한다. 사용자는 원격지에서 인터넷을 통하여 웹 서버에 접속함으로써 자율이동 로봇에게 명령을 내릴 수 있다. 원격운용 시스템의 하드웨어 사양은 자율이동 로봇, 서버 워크스테이션, 로컬 컴퓨터로 구성되어지며 그들은 서로 컴퓨터 네트워크를 통하여 통신이 가능할 수 있도록 고유한 주소(Ethernet address)를 가져야한다. 서버 시스템과 로봇시스템사이의 명령은 무선 랜(wireless LAN)을 통하여 TCP/IP 소켓을 생성하여 전송하고 영상정보는 FTP(file transfer protocol)를 이용하여 전송한다.

2. 로봇 시스템

로봇 시스템은 자율이동 로봇을 운용하기 위한 프로그램으로 영상 압축, TCP/IP 소켓 생성, FTP 서버 연결, 직렬 통신 등을 포함하고 있다. CCD 카메라로부터 입력된 영상을 서버에 전송하기 위해서 정지영상 압축의 표준이 JPEG을 적용하여 그 데이터 양을 감소시켜 파일을 만든 후 파일전송 프로토콜을 이용하여 서버에 있는 홈 디렉토리에 전송한다. 서버에서의 명령은 소켓 포트 1600을 통하여 로봇시스템에 전달되면 이것은 다시 시리얼 포트를 통하여 모터드라이버를 제어하게 된다.

3. 서버 시스템

서버 시스템은 로봇 시스템과 클라이언트 시스템과의 명령을 로봇에게 전송하기 위한 네트워크 프로그램, 전체 원격운용 시스템을 감시하는 모니터링 프로그램 등을 Java 어플리케이션으로서 구현하였다. 서버 시스템의 운영체제는 유닉스를 탑재하고 있으며, 서버 시스템의 모든 프로그램을

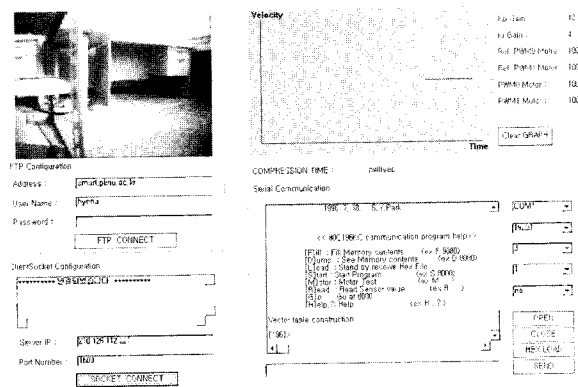


그림 2. 로봇 시스템 인터페이스.
Fig. 2. Graphic interface for robot system.

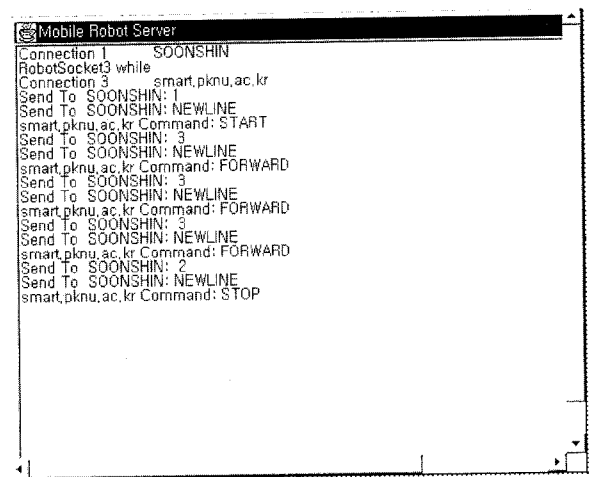


그림 3. 서버 시스템 인터페이스.
Fig. 3. Interface for server system.

Java로 구현하여 다른 운영체제로의 확장성도 고려하였다. 네트워크 프로그램은 무선 랜을 통하여 서버 소켓포트 1500, 1600을 생성하여 클라이언트와 로봇 시스템 사이의 명령을 전송을 하도록 구현하였으며, 시스템의 일관성 있는 명령 체계를 위해서 소켓을 통하여 접속되는 사용자수를 제한하며 로봇 시스템 운용에 관한 권한은 사용자중 한사람만이 가질 수 있도록 하였다. 서버 시스템은 항상 사용자가 접속할 수 있도록 소켓을 생성한 후 쓰레드를 실행하여 대기한다.

4. 클라이언트 시스템

클라이언트 시스템은 영상 디스플레이, 클라이언트 소켓 생성, 명령 버튼 등으로 구성된다. 로봇으로부터 전송된 이미지 정보를 디스플레이 함으로서 사용자는 인터넷을 통하여 원격지의 정보를 얻을 수 있으며, 생성된 소켓 포트 1500을 통하여 서버로 명령을 전송할 수 있도록 구현하였다. 사용자가 브라우저를 통하여 홈페이지에 접속하면 Java 애플릿이 다운로드되어 실행되고 여기서 GUI를 통하여 명령을 내리면 생성된 소켓을 통하여 서버로 전송된다. 클라이언트 시스템을 나타내면 그림 4와 같다.

5. 영상 압축

자율이동 로봇을 실시간으로 운용하기 위해서는 CCD 카메라로부터 입력된 영상데이터 양을 최소화할 수 있는 압축

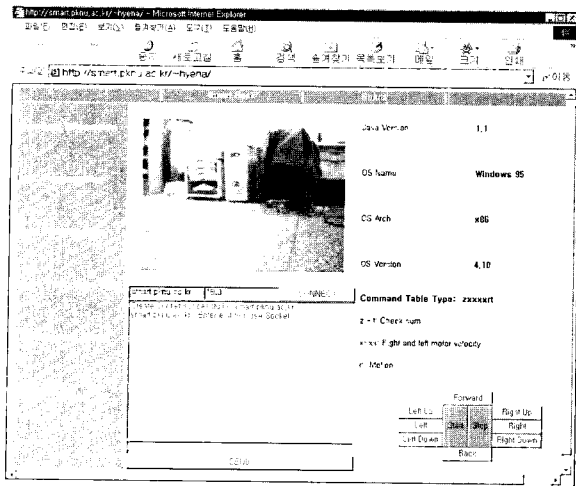


그림 4. 클라이언트 시스템 사용자 인터페이스.
Fig. 4. Graphic user interface for client system.

알고리즘이 우선적으로 필요하다. 이러한 알고리즘은 전체적인 원격운용 시스템의 성능과 직결됨으로, 네트워크 속도와 메인 컨트롤러의 압축 알고리즘 처리 시간을 고려한 가장 최적의 알고리즘을 적용해야만 한다. 본 논문에서는 한 프레임의 영상을 압축하기 위해서 정지영상 압축 알고리즘의 표준인 JPEG을 이용하여 압축하고자 한다[10]. 캡처보드의 영상 데이터는 JPEG압축을 통하여 서버로 전송할 수 있는 파일로 만들어지게 된다.

III. 하드웨어 구성

본 논문에서 제안한 시스템의 하드웨어 사양은 이동로봇, 서버 워크스테이션과 로컬 컴퓨터로 구성되어 진다. 이동로봇은 메인 컨트롤러, 영상처리 시스템, 외장형 무선 랜, 모터 드라이버 보드, DC모터, 초음파센서, 센서 데이터 처리 보드, 충전지 등으로 구성되며, 영상처리 시스템은 CCD 카메라와 캡처보드로 구성된다. 이동로봇은 그림 5와 6에서

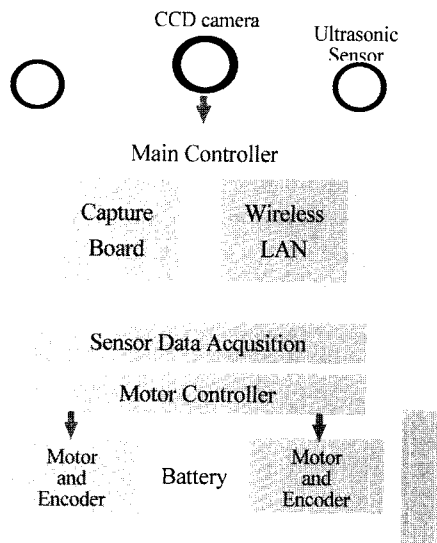


그림 5. 제작된 이동로봇의 전체구조.
Fig. 5. Structure of the developed mobile robot.

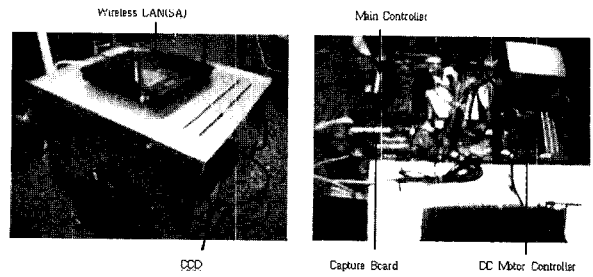


그림 6. 이동로봇의 사진.
Fig. 6. Photographs of the mobile robot.

보는 바와 같이 60cmx45cmx40cm의 크기를 가진다. 또한 Celeron 433MHz CPU를 가진 PC기반으로 설계하였으며, 모든 주변 드라이버와 장치는 메인보드의 CPU에서 관리하게 된다. DC모터 드라이버와 센서의 데이터 처리는 마이크로 컨트롤러를 사용하여 제어하게 되며, 초음파 센서는 정적인 장애물이나 동적인 장애물을 감지하기 위해서 사용한다. 무선 랜을 사용함으로써 로봇은 공간적인 제약을 극복할 수 있으며, 데이터 양이 큰 이미지 전송시간을 줄일 수 있다. 로봇의 무선 랜은 station adapter(SA)를 사용하며 최고 11Mbps 전송률을 가지며 네트워크 카드와 대각선으로 연결된다.

이동로봇은 서버와 FTP 접속, 서버 시스템과 소켓 접속, 시리얼 COM 포터 오픈, 영상보드 세팅 등의 초기화를 마친 후 스타트 명령을 실행할 때까지 대기한다. 원격지 사용자의 명령 테이블을 분석하여 스타트 명령과 일치하면 시리얼을 통하여 모터를 구동한다. 로봇이 동작하는 동안 계속해서 명령 테이블을 분석하여 움직임과 속도를 제어한다. 스톱 명령이 들어오면 로봇은 정지하고 모든 레지스터를 초기화한다. 이것을 나타내면 그림 7과 같다. 로봇의 명령 테이블과 속도 테이블은 7바이트로 구성되며 최상위 바이트와 최하위 바이트는 테이블의 시작과 끝을 나타낸다. 명령 테이블의 나머지 바이트는 로봇의 속도제어 및 움직임을 제어하기 위한 명령으로 사용된다. 속도를 모니터링하기 위한

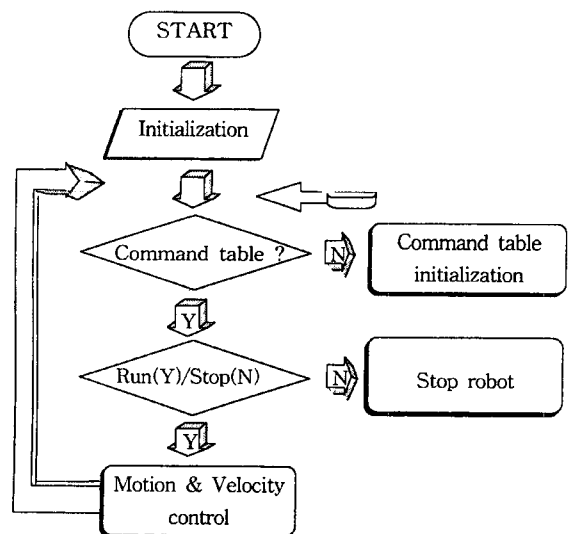


그림 7. 구동 알고리즘.
Fig. 7. Operation algorithm for the mobile robot.

속도데이터는 현재 로봇의 양쪽 모터 속도를 전송하기 위한 데이터 패킷이다.

서버는 Ultima1 250Mhz로 솔라리스 2.6을 탑재하고 있으며 웹 서버와 FTP서버를 설치하였고, 서버의 랜 카드와 외장형 무선 랜 Access Point(AP)는 1:1로 연결한다. 웹 서버로는 유닉스 플랫폼에서 빠르고 안정적이며 세계적으로 가장 보편적으로 사용하는 아파치 서버를 설치하였다.

한편, 이동로봇의 모터제어기로는 80C196KC 프로세스, 모터 드라이브로는 L298, 모터의 속도를 측정하기 위해서는 1024펄스의 엔코더를 사용하였으며 제어 알고리즘으로는 PI속도제어를 수행하도록 설계하였다.

IV. 실험 및 검토

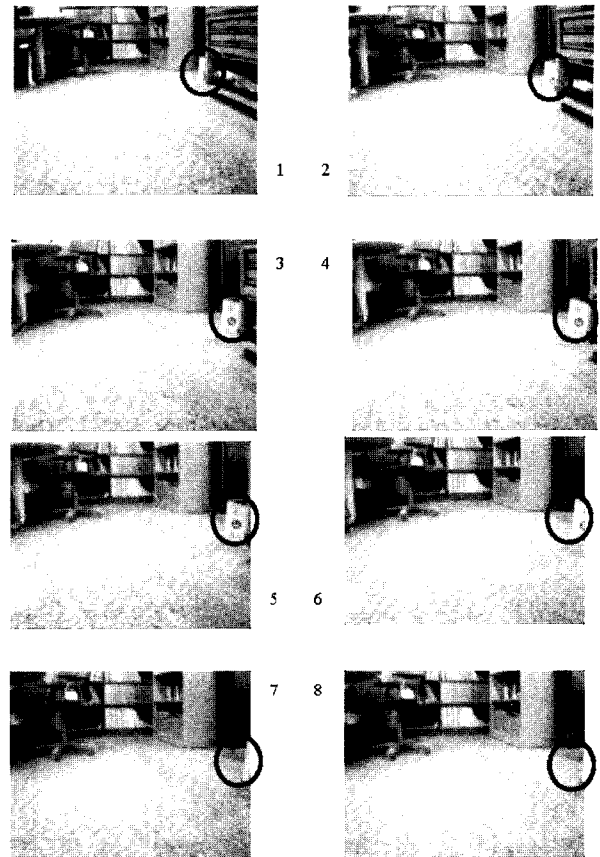
320 X 240 크기의 원 영상은 225Kbyte이며 압축시 1/20 이하로 압축이 가능하며 12Kbyte까지 압축하여 서버에 전송한다. 로봇 시스템에서 한 프레임을 압축하는데 소요되는 시간은 47msec, 압축 파일 업로드 시간은 153msec 가 소요되었다. 1초에 30프레임을 캡처할 수 있는 보드로서 압축만 했을 경우 21프레임, 업로드를 동시에 했을 경우에는 5프레임이 가능하였다.

이동로봇의 원격주행을 위해 이동로봇 속도는 100mm/sec로 제한하였다. 로봇의 이동 반경은 폐쇄적인 환경에서는 65m, 개방된 환경에서는 120m까지 제어가 가능하다. 실험실 내에서 로봇의 주행영상은 그림 8(a)와 같으며, 그림 8(b)는 복도에서 로봇 주행 실험시의 영상 변화를 나타내고 있다.

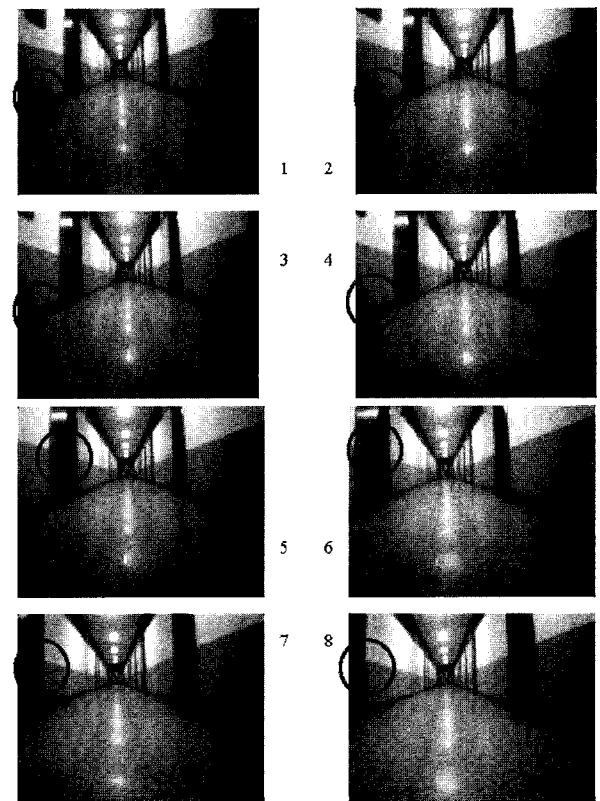
실험에서 볼 수 있듯이 이동 로봇의 주행은 만족할만한 성능을 나타내었다. 그러나 압축 영상을 전송하는 경우에 있어서는 시간이 다소 소모됨을 확인할 수 있었다. 또한 서버와 클라이언트 개념의 원격운용 시스템에 있어서는 Java와 같은 인터넷 지향적이며 플랫폼 이식성이 뛰어난 언어를 사용함으로써 구현과 실험이 용이함을 알 수 있었다. 원격운용시스템을 실시간으로 구현하는데 있어서는 아직 까지 극복해야 할 과제가 많이 남아 있다. 첫 째로 원격지에서 사용자의 명령에 의해 로봇이 동작하므로 하드웨어적 정밀함이 요구되며, 다음으로는 시간 지연으로 인해 현재 영상과 원격지에서 모니터링하는 영상사이에 차이가 발생할 수 있다. 이로 인하여 로봇에게 사용자가 제어할 수 없는 환경변화가 발생할 수 있으므로 이에 대한 대처 능력이 필요하다.

V. 결론

본 논문에서는 새로운 분야로 부각되고 있는 원격운용 시스템을 Java와 컴퓨터 비전을 사용하여 실제로 구현하였으며, 기존의 웹 브라우저를 통하여 원격지의 로봇을 운용할 수 있었다. 인터넷을 통하여 자율이동 로봇을 원격지에서 운용함으로써 기존의 기능에서 탈피한 능동적인 운용이 가능함을 보였다. 그리고 무선 랜을 이용함으로써 운용 공간의 한계를 극복할 수 있었다. 또한 영상 압축 알고리즘을 적용함으로써 사용자에게 이동 로봇으로부터 전송된 원격지의 영상 정보를 실시간으로 공유 할 수 있는 기회를 제공하였으며, 로봇에게 일반 사용자가 한 걸음 더 다가갈 수 있는 계기를 마련하였다. 그러나 실험에서 보듯이 네트워크



(a) 실험실 내



(b) 복도

그림 8. 클라이언트로 전송된 로봇 이동중의 영상 변화.
Fig. 8. Images transmitted from the mobile robot to a client.

망의 병목현상으로 인한 시간지연은 피할 수 없는 현상으로 네트워크 속도가 시스템의 성능과 직결됨을 알 수 있었다. 그러나 네트워크 속도가 개선된다면 완전한 실시간 운용이 가능함을 확인할 수 있었고, 이 경우 이동로봇의 원격운용 시스템은 다양한 분야에서 적용이 가능할 것이다.

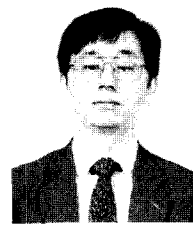
참고문헌

- [1] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, *Desktop Int. Conf. Robotics and Automation*, pp. 654-659, 1995.
- [2] E. Paulos and J. Canmy, "Delivering real reality to the world wide web via telerobotics," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Mineapolis, MN, pp. 1694-1699, 1996.
- [3] J. E. F. Baruch and M. J. Cox, "Remote control and robots: An Internet solution," *IEE Computing Contr. Eng. J.*, pp. 39-44, 1996.
- [4] P. Saucy and F. Mondada, "KhepOnTheWeb: One year of access to a mobile robot on the Internet," *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 22-30, 1998.
- [5] K. S. Tso, P. G. Backes, and G. K. Tharp, "Mars pathfinder mission internet-based operations using WITS," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, pp. 284-291, 1998.
- [6] Dirk Schulz, Wolfram Burgard, Dieter Fox, Sebastian Thrun, and Armin B. Cremers, "Web interfaces for mobile robots in public places," *IEEE Robotics & Automation Magazine*, pp. 48-56, March, 2000.
- [7] <http://java.sun.com>.
- [8] Marry Campione and Kathy Walrath, "The Java™ tutorial second edition: object-oriented programming for the Internet," Addison-Wesley Publishing Company, 1998.
- [9] David Flanagan, "Java in a nutshell," *O'Reilly and Associates*, Third Edition, 1999.
- [10] Rafael C. Gonzalez and Richard E. Wood, "Digital image processing," Addison-Wesley Publishing Company, 1993.



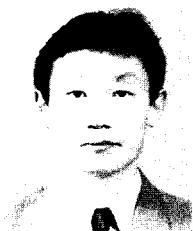
박 태 현

1971년 2월 17일생. 1998년 부경대 제어계측공학과 졸업. 동대학 대학원 전자공학과 석사(2001). 2001년~현재 (주)에이앤에스 연구원. 관심분야는 로보틱스, 영상처리.



강 근 택

1953년 3월 12일생. 1976년 서울대 응용물리과 졸업. 동경공업대 시스템과학전공 석사(1984). 동대학 박사(1987). 1987년~현재 부경대학교 전자컴퓨터정보통신공학부 교수. 관심분야는 퍼지시스템, 지능시스템.



이 원 창

1960년 10월 2일생. 1983년 서울대 제어계측과 졸업. KAIST 전기 및 전자공학과 석사(1985). 포항공과대 전자전기공학과 박사(1992). 1993년~현재 부경대학교 전자컴퓨터정보통신공학부 부교수. 관심분야는 로보틱스, 지능 시스템, 임베디드 시스템.