

빌딩청소용 이동로봇을 위한 원격 모니터링 시스템

Remote Monitoring System for a Building Cleaning Mobile Robot

이 수 영¹, 조 원 호², 최 병 옥[†]

Soo-Yeong Yi¹ · Won-Ho Cho² · Byoung-Wook Choi[†]

Abstract This paper presents a remote monitoring and simulation system for a building cleaning mobile robot. It provides a tool of convenient 3D graphical map construction including network camera image viewer and status information of the robot. The 3D map is reconstructed from existing 2D building CAD data with DXF format using OpenGL graphic API. Through this system, it is possible to monitor and control the cleaning mobile robot from remote place. A practical experiment is performed to show the reliability and convenience of the monitoring system. The proposed system is expected to give efficient way of control and monitoring to building cleaning mobile robot.

Keywords: 3D map building, image viewer, remote control

1. 서론

지능형 서비스 로봇은 다양한 첨단 기술의 융합체로서, 국내외적으로 차세대 성장 산업으로 지정되어 국가적인 연구개발과 산업화를 추진하고 있다^[1]. 원격지에서의 실시간 제어 및 모니터링 기술은 원격 네트워크로 연결된 로봇의 다양한 정보를 제공하여 사용자가 실시간으로 로봇 상태를 모니터링하고 상황에 맞는 명령을 빠르게 적용하려는 연구이다. 이는 서비스 로봇의 부가 가치를 높이는 것으로 시간과 지역에 독립적이라는 특징이 있어서 많이 활용되고 있다^[2]. 본 논문에서는 개발 중인 빌딩청소용 이동 로봇을 위한 원격 모니터링 시스템을 구현하였다.

마이크로소프트에서 제공하는 로봇 통합 개발 환경인 MSRDS(Microsoft Robotics Developer Studio)를 이용하여 시뮬레이션 환경을 구성하거나, 실세계 모니터링을 위한 다양한 클래스를 제공하는 Java3D API를 이용하여 로봇의 가상 실험 환경을 구현한 연구가 발표되었다^{[3][4]}. MSRDS의 경우 시뮬레이션 환경을 통해 하드웨어 로봇과 소프트웨어 개발의 병행이 가능하도록 하고, 소프트웨어

를 테스트 할 수 있는 시뮬레이션 환경을 제공함으로써 비용과 시간을 감소시키는 효과를 얻을 수 있다. 그러나 서비스의 분산은 전체 프로그램을 이해하는데 어려움을 주고 가상 로봇의 모션제어 역시 아직은 미흡한 상태이다^[5]. 또한 .Net Framework나 JVM(Java Virtual Machine)을 설치해야 하므로 개발 환경에 부담이 될 수 있다.

본 논문은 Visual C++ MFC를 개발 도구로 사용하고, 3차원 그래픽 라이브러리인 OpenGL API를 사용하여 구현한 빌딩청소용 이동로봇을 위한 원격 모니터링 시스템을 기술한다. 이는 많은 사용자를 확보한 도구를 이용하여 최적화된 환경 구축이 가능한 이점이 있다.

원격 모니터링 시스템은 크게 3D 맵 빌딩 및 시뮬레이션 부분, 네트워크 영상과 로봇 상태 모니터링 부분, 그리고 그래픽 사용자 인터페이스(GUI, Graphic User Interface)를 통한 로봇의 원격조작부분으로 구성된다. 본 논문은 원격 네트워크 모니터링 시스템을 통해 빌딩청소용 로봇의 활용 범위를 넓히는데 목적이 있다.

본 논문의 구성은 다음과 같이 구성하였다. 2장에서 빌딩용 청소로봇을 위한 모니터링 시스템의 구조 및 기능을 소개하고, 3장은 실험 결과, 그리고 4장에서 결론을 맺도록 한다.

Received : February 2nd, 2009 Accpeted: February 20th, 2009

† 교신저자 : 서울산업대학교 전기공학과 부교수

(E-mail : bwchoi@snut.ac.kr)

¹ 서울산업대학교 전기공학과 부교수(E-mail : suylee@snut.ac.kr)

² 서울산업대학교 전기공학과 석사과정(E-mail : cwh0730@snut.ac.kr)

2. 빌딩용 청소로봇 모니터링 시스템

2.1 전체 시스템 구조

전체 모니터링 시스템은 그림 1과 같이 3차원 맵 빌딩, 영상 뷰어/로봇 상태 모니터링 그리고 그래픽 인터페이스를 통한 로봇의 원격 조작 등 크게 세부분으로 구성된다.

본 시스템에서 로봇의 청소구간의 맵은 2차원 도면파일(DXF 파일)로 제공된다. 2차원 도면은 3차원 그래픽 라이브러리인 OpenGL API를 사용하여 3차원 화면으로 재구성하여 가상공간을 만든다. 로봇모델 역시 OpenGL API를 이용하여 렌더링하고 3차원 물체는 직육면체 형태로 단순화하여 표현한다. 각 물체와 로봇 정보는 파일 처리를 통해 편의성 및 확장성을 추구하였다.

다음으로 영상 뷰어부분은 로봇에 장착된 CMOS 카메라를 통해 얻은 영상 데이터를 JPEG 코덱을 이용하여 TCP/IP 네트워크 통신으로 원격 모니터링 프로그램에 전송, 나타낸다. 그 외에도 로봇의 배터리 잔여 전력량과 각종 센서 데이터, 범퍼, 긴급 상황 등을 표시하여 로봇의 상태를 감시한다.

마지막으로 그래픽 버튼 인터페이스를 이용한 네트워크 원격조작으로 로봇을 전진, 후진, 좌우 회전으로 제어하여 청소로봇의 활용성을 높였다. 차동 바퀴형 이동로봇의 기구학을 적용하여 모니터링 프로그램에 반영함으로써 좀 더 사실적인 로봇의 청소 모습을 보이고자 하였다.

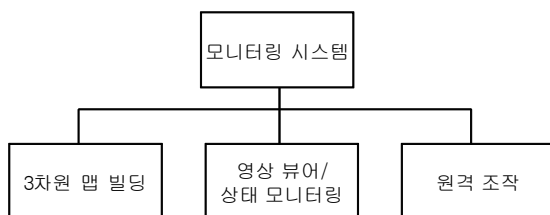


그림 1. 모니터링 시스템 구조

2.2 청소 환경에 대한 3차원 맵 구축

청소 영역에 대한 3차원 맵 구축은 원격지에 있는 사용자가 로봇 주변의 온라인 상황을 입체적으로 실감나게 파악하고, 장애물이 있는 환경에서 로봇의 청소주행을 미리 오프라인 시뮬레이션을 통해 확인할 수 있게 하기 위한 목적이 있다.

3차원 맵을 구축하는 방법은 크게 세가지 방법으로 분류된다. 먼저 사용자가 측정기를 이용하여 장애물 위치 및 크기를 손수 측정하고 기록하는 방법이 있다. 이는 맵이 바뀔 때마다 다시 수작업으로 일일이 작성해야 하므로 매우 번거롭고 확장성이 없다.

다음으로 이동로봇이 자신에게 장착된 거리 센서를 이

용하여 스스로 장애물 환경을 구축하는 방식이 있다. 그러나 이는 정확한 지도를 작성하기 위해서는 로봇의 위치추정이 정확하여야 하지만, 정확한 위치추정을 위해서는 지도가 정확하여야 하는 등 두 문제가 서로 얽혀 있기 때문에 오랫동안 연구에도 불구하고 완전한 해법이 제시되지 못하고 있다.

마지막으로 미리 주어진 빌딩 내부 2차원 맵 도면을 인터페이스하는 방식으로 본 논문에서는 2차원 CAD도면 작업을 인터페이스하여 3차원 맵을 작성한다.

로봇 주변환경에 대한 3차원 맵을 구축하기 위한 과정은 다음과 같다.

- 기존의 건물내부 2차원 도면 DXF 파일 적재
- 적재된 도면을 OpenGL API를 이용, 3차원 구성
- 로봇과 각 오브젝트 모델들을 렌더링

2.2.1 건물 내부의 2차원 CAD 도면 인터페이스

청소할 맵의 확장성과 구축과정의 편의성을 위해 미리 주어진 건물내부 2차원 CAD 도면 DXF(Drawing eXchange File) 파일을 모니터링 시스템에 인터페이스하였다. DXF는 미국 오토데스크사의 AutoCAD와 다른 여러 종류의 CAD 소프트웨어들 간의 파일 교환을 위한 표준 포맷으로 맵에 대한 도면을 만드는데 사용된다.

그림 2는 DXF파일을 인터페이스하는 과정의 순서도이다. DXF파일은 그래픽과 관련된 변수의 설정 값을 갖는 HEADER 섹션, 공간실체(entity) 변수 개수를 갖는 TABLE 섹션, 각각의 블록의 entry를 포함하는 BLOCK 섹션, 도면을 형성하는 공간 실체를 저장하는 ENTITY 섹션 등 크게 4가지 섹션으로 구성된다⁶⁾.

본 논문에서는 DXF 파일의 HEADER 섹션에서 도면 공간의 크기를 얻어오고 ENTITY 섹션의 점, 선, 원, 아크, 텍스트 등을 구별하여 메타파일(Metafile) 형태로 저장해 두었다가 화면에 표시한다. 메타파일은 그래픽 이미지를 생성해낼 수 있는 GDI(Graphic Device Interface) 명령들로 구

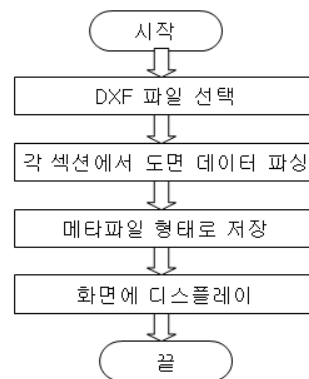


그림 2. DXF파일 인터페이스 과정

성된 파일로써 일반 이미지 파일에 비해 크기가 작고 축소나 확대 시에도 비트맵의 손실 이 거의 일어나지 않는다^[7].

그림 3은 DXF 파일을 인터페이스하여 화면에 나타낸 결과를 보여준다. 2차원 도면이 적재되면 이를 3차원 그래픽 라이브러리인 OpenGL API를 사용하여 3차원 데이터로 변환한다. 화면 픽셀의 RGB값을 배열에 저장한 후 이를 바탕으로 X, Y, Z축의 3차원 화면으로 그림 4와 같이 재구성한다.

OpenGL API를 이용하여 2차원 컴퓨터 화면에서 3차원 도면을 효과적으로 표현하기 위해서는 다음 4가지 과정을 거쳐야 한다.

- 뷰잉 변환 (Viewing Transformation)
- 모델링 변환 (Modeling Transformation)
- 투영 변환 (Projection Transformation)
- 뷰포트 변환 (Viewport Transformation)

맵의 표현은 3차원 공간의 내용을 이차원적인 화면에 표시하기 위해 직교 투영(Orthographic Projection)방식을 사용한다. 이는 영역 내에 있는 모든 물체들이 거리에 관계없이 모두 같은 크기 비율로 표현되는 방식으로 흔히 건축물의 설계도나 디자인 문서 등에 사용된다. 렌더링에 필요한 3D 공간을 만든 다음, 모델뷰(Modelview) 변환과 뷰포트(Viewport) 변환을 통해서 맵 구성에 필요한 지오메트리

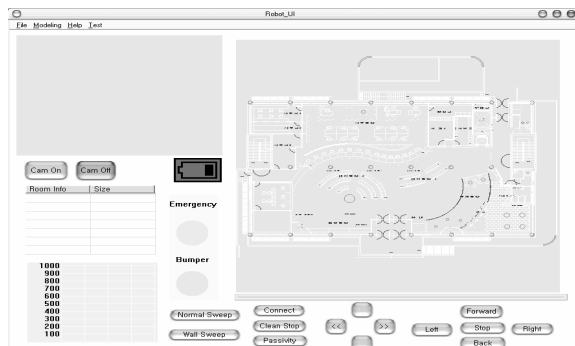


그림 3. 기존 2차원 도면 인터페이스

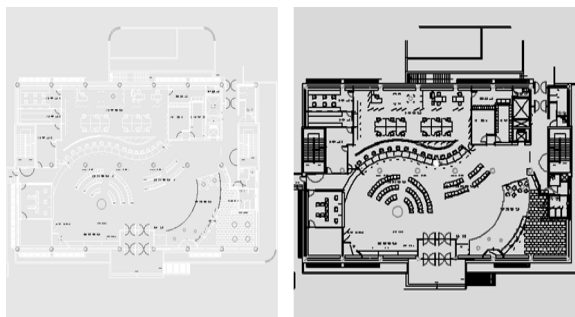


그림 4. 2차원 화면을 3차원으로 재구성 (Z축이 0)

(Geometry)들을 렌더링한다^[8]. 그래픽에서 모든 기하변환은 행렬의 곱셈 형태로 구현되는데 다음 식과 같은 여러 가지 행렬 연산으로 3차원 맵의 시각을 다양하게 연출할 수 있다.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (2)$$

식 (1)은 Z축을 기준으로 반 시계방향으로 θ 만큼 회전하는 변환행렬식이고 식 (2)의 S_x, S_y, S_z 는 X, Y, Z축 방향의 배율로서 배율이 1보다 크면, 확대, 작으면 축소에 해당하고 그것에 맞게 크기를 조절하는 변환행렬식이다. 그림 5는 회전 행렬변환을 통해 맵의 시각을 변경하여 화면을 다양한 각도로 변환한 결과를 보여준다. 그림 6은 크기 조정 행렬변환을 통해 화면을 확대, 축소한 것이다.

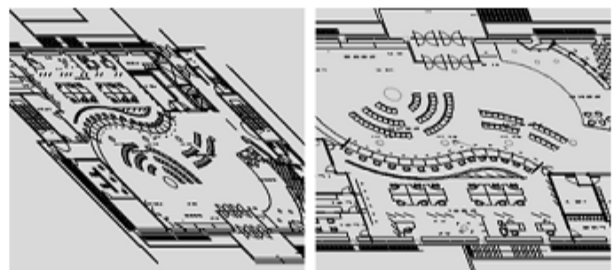


그림 5. 회전 행렬변환을 통한 3차원 시각 설정

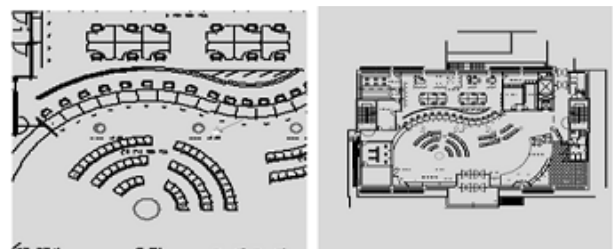


그림 6. 크기 조정 행렬변환을 통한 확대 축소

2.2.2 3차원 장애물 및 로봇 모델 렌더링

3차원 맵 공간을 만든 다음, 로봇을 포함한 빌딩 내부공간의 3차원 물체들을 사용자에게 실감 있게 제시하기 위해 물체에 대한 렌더링이 필요하다. 모든 3차원 물체는 여러 개의 다각형 집합으로 표현된다. 본 논문에서는 모델의

단순화를 위하여 벽, 장애물 등 빌딩내부 공간의 3차원 물체 렌더링을 위한 표현 단위를 직육면체로 가정하여 렌더링하고, 이동로봇은 3차원 CAD 설계도면을 그대로 인터페이스하여 렌더링하는 방식으로 구현하였다.

빌딩내부 공간의 물체들을 표현하는 직육면체는 6개의 사각형 표면 메쉬로 표현할 수 있다. 각 직육면체는 전역 좌표계에서의 x, y, z 위치와 Z 축을 기준으로 한 회전방향, 그리고 폭, 높이, 깊이 정보를 가지고 있다. 복잡한 모양의 오브젝트를 직육면체로 단순화하거나, 혹은 크고 작은 여러 개의 직육면체들을 조합하여 보다 정교하게 표현할 수 있다. 3차원 물체는 그림 7과 같이 모달리스(Modeless) 다이얼로그에서 좌표 설정을 통해 생성하며, 텍스트 파일을 통하여 오브젝트들과 로봇의 정보를 저장/적재함으로써 3차원 맵의 재사용을 가능하게 하였다.

청소 로봇 모델은 3차원 모델링 CAD 소프트웨어인 CATIA의 stl파일을 불러와서, 로봇을 구성하는 물체들의 표면 메쉬를 나타내기 위한 꼭지점(vertex)들의 배열로 저장된다. 로봇 표면의 곡면과 같은 복잡한 형태를 높은 정밀도로 표현하기 위해서는 일반적인 CAD 소프트웨어에서는 삼각형 표면 메쉬를 사용한다⁹⁾.

로봇은 상단, 중간, 하단 프레임, 바퀴와 같은 각 부품 물체의 결합으로 이루어지고, 이를 OpenGL API인 glArrayElement() 함수를 사용하여 그림 8과 같이 렌더링한다¹⁰⁾.

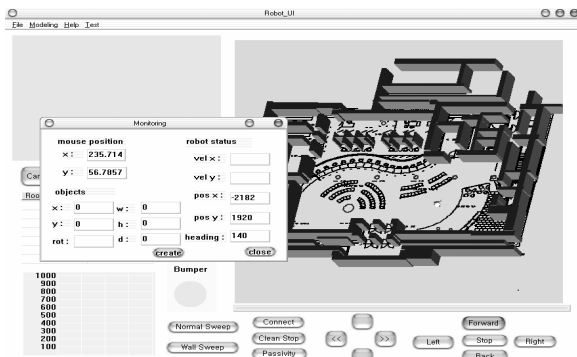


그림 7. 3차원 맵 빌딩

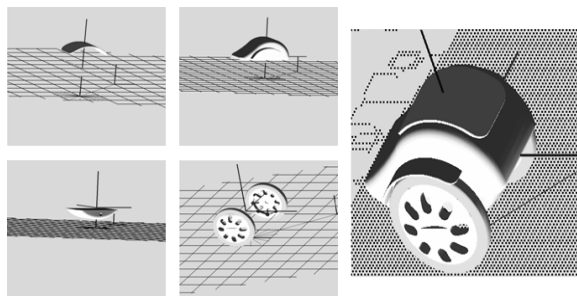


그림 8. 3차원 로봇 모델의 렌더링

2.3 네트워크 영상과 로봇 상태 모니터링

본 논문의 원격 모니터링 시스템은 영상이나 로봇의 상태를 실시간으로 보여준다. 원격지에서도 사용자가 실제 환경이나 로봇의 정보를 확인하고, 그것에 적절한 대응을 함으로써 청소 로봇의 활용 범위를 확대할 수 있다.

영상 모니터링은 로봇에 장착된 CMOS 카메라를 통해 얻은 영상 데이터를 JPEG 코덱을 이용하여 압축하고, 이를 TCP/IP 네트워크를 통해 전송하는 방식으로 구현 하였다¹¹⁾.

로봇 주변 청소공간의 영상 모니터링 외에 로봇의 상태를 모니터링하기 위해서 배터리 잔여 전력량과 각종 센서 데이터, 범퍼 그리고 긴급 상황에 대한 정보를 모니터링할 수 있도록 하였으며, 이를 위해 그림 9와 같이 네트워크 데이터 패킷을 정의하였다.

하나의 패킷은 STX, ETX를 포함하여 26bytes이고 배터리, 범퍼의 눌림 상태, 긴급 상황 유무, 로봇의 좌표와 방향각 순서로 구성된다. 네트워크 통신을 위해서 영상전송을 위한 쓰레드와 상태정보 전송을 위한 쓰레드를 생성하여 병렬 처리를 한다. 전송된 로봇 상태 정보들은 모니터링 시스템상에 그림 10과 같이 그래픽으로 표현된다. 배터리 잔여 전력량은 3단계로 나누어서 GUI 형태로 나타내고, 각종 센서 데이터는 그래프로 나타내어 변화를 감지한다. 그리고 로봇 범퍼의 눌림이나 긴급 상황 발생에 대한 감지는 그라디언트(Gradient) LED로 표현하였다.

STX	battery	bumper	emergency	로봇 x 좌표	로봇 y 좌표	전면 각도	ETX
-----	---------	--------	-----------	---------	---------	-------	-----

그림 9. 로봇의 상태정보 전송을 위한 패킷 프로토콜

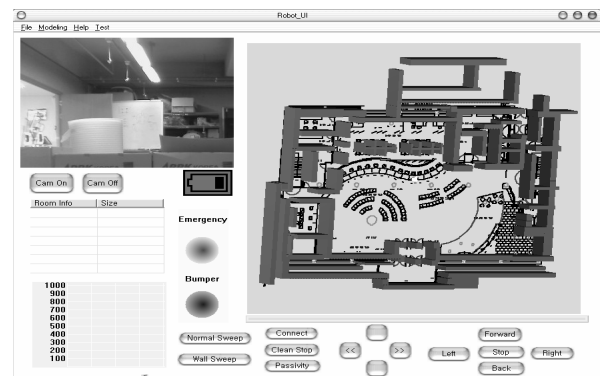


그림 10. 네트워크 영상과 로봇 상태 모니터링

2.4 로봇의 주행 시뮬레이션 및 원격 제어

본 모니터링 시스템은 차동 바퀴 이동로봇의 기구학을 적용하여 오프라인으로 로봇의 이동을 시뮬레이션 할 수 있다. 차동 바퀴형의 이동로봇은 나란한 두 개의 구동 바퀴와 로봇 몸체를 지탱해 줄 1~2개의 보조바퀴로 구성되며 두 개의 구동 바퀴를 이용해 자율 이동이 가능한 로봇

이다^[12].

그림 11은 차동 바퀴 이동 로봇의 기구학을 나타낸다. 현재 위치 (x, y, θ) 에서 로봇의 몸체 속도를 v , 방향 각 속도를 w 라 하였을 때 이 몸체 속도 및 방향 각속도는 양 바퀴에 대한 선속도명령으로 나타낼 수 있다. 단순 차동형 바퀴를 갖는 이동로봇의 몸체 및 바퀴의 속도 관계식은 식 (3)과 같다.

$$w = \frac{V_R - V_L}{L}, \quad v = \frac{V_R + V_L}{2} \quad (3)$$

$(L : \text{바퀴의 반지름})$

이를 바탕으로 이동 속도 (x_c, y_c, θ_c) 를 나타내면 식 (4)와 같다^[13].

$$\begin{pmatrix} x_c \\ y_c \\ \theta_c \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (4)$$

원격조작 기능은 청소의 시작, 종료 명령뿐만 아니라 청소가 미진한 부분을 좀 더 청소하거나 청소상태를 국부적으로 모니터링하는등 활용 범위가 넓다. 사용자 인터페이스 버튼을 누르면 전진, 후진, 좌우회전 명령을 내리는 신호를 전송하기 위한 프레임이 생성되어 TCP/IP 통신으로 로봇에 전송한다. 그림 12는 명령전송을 위한 패킷 프로토콜이다.

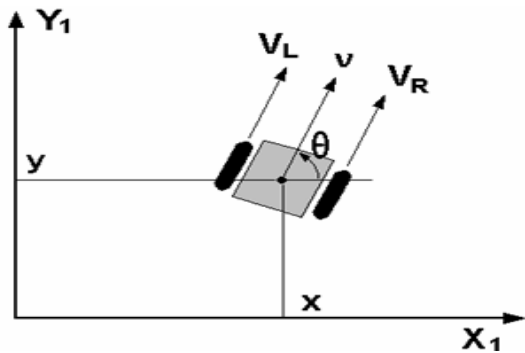


그림 11. 차동 바퀴 이동로봇의 기구학

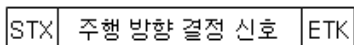


그림 12. 원격제어 명령전송을 위한 패킷 프로토콜

3. 실험 결과

본 논문에서 개발한 원격 모니터링 시스템을 연동 실험하기 위해 (주)다사 로봇에서 개발한 빌딩용 청소로봇

ZACSEN2를 사용하였다^[14]. ZACSEN2는 지식경제부 국책 과제로 개발된 국내 최초 빌딩용 청소 로봇으로, 950x1600mm 크기에 0.5m/sec의 최대 속도로 갖는다. 그림 13은 빌딩청소용 로봇과 모니터링 시스템간의 운영 시나리오를 나타낸다. TCP/IP 프로토콜을 이용한 네트워크 통신으로 로봇은 서버, 모니터링 시스템이 클라이언트가 되어 통신한다.

네트워크 접속 방식은 시스템의 부하를 줄이기 위해 지속적인 접근이 아닌 데이터 필요시 접속과 해제를 반복하는 방식을 택하였다. 로봇으로부터 받은 패킷 정보에는 청소로봇의 x, y 위치 및 방향각(θ)이 포함되어 있으며, 이로부터 사용자가 청소로봇이 빌딩내부 공간의 어느 위치에 있는지 실시으로 알 수 있다. 물론 전술한 바와 같이 배터리나 범퍼, 긴급 상황에 대한 정보 등 로봇의 상태도 실시간으로 모니터링 할 수 있다. 로봇의 주행은 격자 주행과 벽면주행으로 나누어 실험하였다. 격자 주행은 위치정보 센서를 이용하여 청소 공간을 격자 모양으로 나누어서 지도 내부부를 지그재그 모양으로 주행하는 것이고, 벽면 주행은 초음파 센서 등의 거리 센서를 사용하여 로봇과 벽 사이의 일정한 거리를 유지하면서 벽을 따라 주행하는 것이다.

다음 그림 14, 그림 15는 청소 로봇이 벽면 주행을 하고

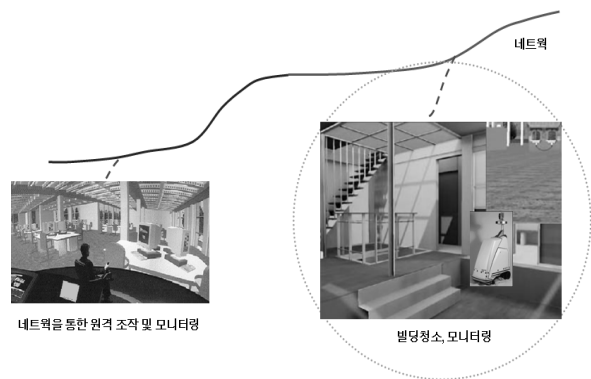


그림 13. 빌딩청소용 로봇 운영 시나리오



그림 14. 빌딩청소용 이동로봇의 벽면 주행

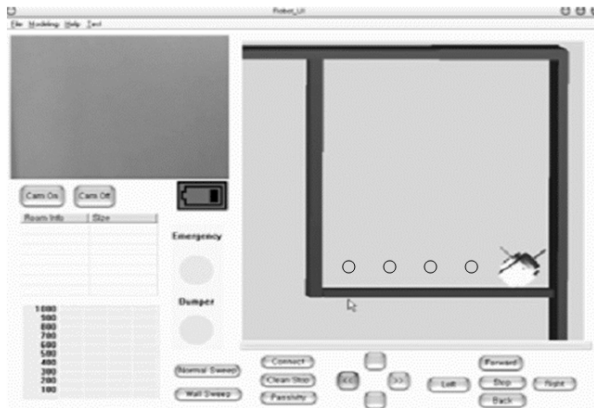


그림 15. 모니터링 시스템

있을 때 모니터링 시스템에서 이를 확인하는 모습이다. 로봇의 그래픽 표현은 TCP/IP 네트워크 통신을 통해 50ms 주기로 받은 로봇의 좌표 및 방향각 데이터 값을 이용하여 그림을 지속적으로 업데이트시켜 가상 환경의 모습을 렌더링한다.

그림 15의 하단부에는 앞에서 언급한 로봇 원격조작을 위한 명령 버튼을 마련되어 있다. 원격조작 버튼들은 전진, 후진, 좌/우 회전, 정지 버튼으로 구성되어 있으며 각 버튼을 누를 때마다 쓰레드를 생성하여 로봇에게 그에 맞는 신호를 네트워크 통신으로 전달함으로써 원격제어를 한다.

4. 결론

로봇의 시뮬레이션이나 모니터링 시스템은 컴퓨터의 가상 모드로 실제계의 환경이나 로봇의 움직임 경로를 나타내어 원격지에서라도 최적의 일을 수행하도록 한다. 또한 로봇의 작업 수행 과정을 관찰하여 사용자가 상황에 맞는 명령을 내려 잘못된 작업을 수행하는 것을 미연에 방지할 수 있다.

본 논문에서는 빌딩청소용 이동로봇을 위한 네트워크 모니터링 시스템을 구현하여 시간과 공간에 구애받지 않고 실시간으로 로봇의 정보를 모니터링 함으로써 좀 더 효율적인 청소 로봇의 활용 방안을 제시했다.

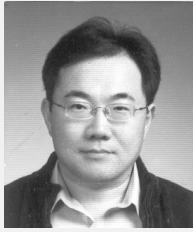
로봇의 주변 환경 맵 구축을 위해서 기존의 2차원 빌딩 CAD 도면을 인터페이스한 후 사전에 청소주행 시뮬레이션을 하였다. 그리고 가상공간에 머무는 것이 아니라 실제 로봇과 연동하여 실험하고 원격조작을 통해 청소 로봇의 활용 범위를 넓혔다.

본 연구의 향후 과제로 DXF파일 맵을 적재할 때 공간 실체(entity) 정보를 이용하여 자동으로 3차원 물체가 생성되도록 하여 사용자가 3차원 형태의 외곽(벽)을 일일이 생성해야 하는 번거로움을 줄일 예정이다. 그리고 기존의 직

육면체 형태의 오브젝트 표현 방식에서 나아가 블록, 실린더 등 물체 표현요소(primitive) 확장을 통한 정교한 물체 모델링들을 표현하여 좀 더 현실적인 맵을 구축 할 계획이다. 또한 구획별 청소를 위한 색선 구분하여 청소 구간을 정하고, 원격조작을 위한 조이스틱 인터페이스를 추가한다면 사용자가 더욱 효율적으로 활용할 수 있을 것이다.

참고문헌

- [1] 최병욱, 박광현, 홍지만, 이관우, 서일홍 “네트워크 로봇을 위한 통합 소프트웨어 플랫폼”, 한국로봇공학회 학회지, pp.20-27, 제5권, 제1호, 2008.
- [2] 김주만, “URC 로봇 원격 모니터링기술 개발”, 한국콘텐츠학회 논문지, pp.8-19, 제6권, 제8호, 2006.
- [3] Wei-Han Hung, Peter Liu, and Shih-Chung Kang, “Service-Based Simulator for Security Robot”, IEEE International Conference on Advanced Robotics and its Social Impacts, Aug. 23-25, 2008.
- [4] Du Zhijiang, Yang Donghai, Sun Lining and Fu Lixin, “Virtual Robot Simulation and Monitoring System Based on Java3D”, IEEE International Conference on Robotics and Biomimetics, Aug. 22-26, 2004.
- [5] 신동관, 이성훈, 이수영, 최병욱, “MSRS 시뮬레이션 환경에서의 가상 로봇의 네트워크제어”, 한국로봇공학회 논문지, pp.242-248, 제 2권, 제 3호, 2007.
- [6] Autodesk DXF Reference Guide.
- [7] Visual C++ 완벽가이드 2nd Edition, 김용성, 영진닷컴.
- [8] OpenGL Super Bible 제 3판, Richard S Wnigt, 정보문화사.
- [9] OpenGL로 배우는 컴퓨터 그래픽스, 주우석, 한빛미디어.
- [10] OpenGL 프로그래밍 가이드 제4판, Dave Shreiner, 정보문화사.
- [11] 코텍의 세계로의 초대, 박기현, 홍릉과학출판사.
- [12] 강재구, 이중재, 지민석, 유범재, “퍼지 위험지수에 의한 이동로봇의 물체 추적 및 장애물 회피 주행 제어기”, 한국로봇공학회 논문지, pp.212-220, 제2권, 제3호, 2007.
- [13] 이영민, “PXA270을 이용한 임베디드 리눅스 기반의 경비로봇 주행시스템”, 서울산업대학교 전기공학과 대학원, 2007.
- [14] (주)다사 로봇 홈페이지: <http://www.dasarobot.com>



이수영

- 1988 연세대학교 공과대학 전자공학과(학사)
- 1990 한국과학기술원 전기 및 전자공학과(공학석사)
- 1994 한국과학기술원 전기 및 전자공학과(공학박사)

1995~1999 한국과학기술연구원 휴먼로봇연구센터 선임연구원

1999~2007 전북대학교 전자정보공학부 부교수
2007~현재 서울산업대학교, 전기공학과 부교수
관심분야 : Walking robot system, Gait design and motion control, Robot vision



최병욱

- 1986 한국항공대학교 전자공학과(학사)
- 1988 한국과학기술원 전기 및 전자공학과(공학석사)
- 1992 한국과학기술원 전기 및 전자공학과(공학박사)

1988~2000 LG 산전주식회사, 엘리베이터 연구실장 및 임베디드 시스템 연구팀장

2000~2005 선문대학교, 제어계측공학과 부교수
2005~현재 서울산업대학교, 전기공학과 부교수
관심분야 : 임베디드 시스템, 실시간 제어 시스템, 지능형 로봇



조원호

- 2008 서울산업대학교 전기공학과(학사)
 - 2008~현재 서울산업대학교 전기공학과 석사과정
- 관심분야 : 지능형 로봇, 임베디드 SW, 컴퓨터 비전