

가상 환경에서의 힘 되먹임을 이용한 지능적 조립 시스템: 가상 접촉 검사 및 가상 힘의 계산

An Intelligent Assembly System using Force Feedback in the Virtual Environment: detection of virtual-contact and computation of virtual-contact force

최종오, 김형욱*, 서일홍*, 이병주**

*한양대학교 전자공학과 (Tel : +82-345-408-5802; Fax : +82-345-408-5803; E-mail : jochoi@scorpio.hanyang.ac.kr)

**한양대학교 제어계측공학과 (Tel : +82-345-400-4081; Fax : +82-345-408-5803; E-mail : bj@hyunpl.hanyang.ac.kr)

Abstract : This paper discusses a method of computing the interaction forces produced between objects that will contact in virtual environments. Our previous virtual assembly system is made with 3D graphics techniques. To display constraint forces that will occur during teaching and assembly tasks in virtual environments. A general algorithm is proposed. To show the validity of the proposed method, several simulation results illustrated.

Keywords : force feedback, virtual environment, contact point, haptics, assembly system

1. 서론

가상현실이란 인간의 오감에 신호를 전달하여 실재하지 않는 환경이나 먼 거리에 있는 환경을 현재 그 환경에 실재한다는 착각을 일으켜 상호작용을 가능하게 하는 기술이다. 따라서 가상현실 시스템에서는 사용자에게 자연스럽게 편리한 상호작용을 허락하는 것이 현실감을 극대화하는데 매우 중요한 요소가 되고, 이는 사용자의 행위전달이 자유로워야 하며 가상세계에 실재하고 있다는 착각을 일으키게 할 힘 되먹임과 적절한 감각 생성이 제공되어야 함을 의미한다. 가상현실 시스템에서 감각 생성을 위해 고안된 인터페이스 장치들은 시각 표시 장치, 위치 추적 장치, 삼감형 입력 장치, haptic (촉감이나 힘 되먹임이 가능한) 인터페이스 장치에 대한 개발이 많이 뒤떨어져 있다.

Haptic 이라는 단어는 그리스어로 '만지다' 을 의미하는 말에서 유래한 것으로서 haptic 감각이란 손을 사용하여 느끼는 모든 감각을 의미한다. 인간의 haptic 감각 정보는 크게 촉각 정보와 운동 감각 정보로 나뉠 수 있다. 이러한 인터페이스는 초기에 원격 작업이나 장애인들을 위한 장치로 개발되어 왔는데 현재는 광범위한 사용이 인식되면서 연구 개발이 활발하게 이루어지고 있다[1,6].

산업 현장에서 조립 로봇을 이용한 작업을 수행하는 데 있어서 가장 어려운 점 중에 하나가 다른 일을 시키기 위한 프로그램을 작성하는 것이다. 특히 그 중에서 조립 점에 대한 위치를 알아내기 위한 교시 작업이 가장 번거롭고 비용이 많이 드는 작업이다. 이러한 문제점을 해결하기 위한 방법이 조작자가 가상 환경에서 부품들을 조립하면 가상 환경 조립 시스템이 교시점을 생성하여 로봇 프로그램을 만드는 것이다. 하지만 가상 환경 속에서 조작자가 조립을 행할 때 부품들의 상호 작용 (충돌, 접촉 등)을 전달해 주지 않으면 가상 환경 조립 시스템을 사용하는

데 상당한 불편한 점이 발생하게 된다. 이러한 점을 해결하기 위한 방법으로 가상 환경 조립 시스템에 haptic 인터페이스를 추가하는 것이다.

따라서, 본 논문에서는 공장에서 많이 쓰이는 조립 로봇을 대상으로, 조립 로봇에 다른 일을 수행시키기 위한 프로그램시 사용자가 직접 로봇을 움직여서 조립 위치에 대한 정보들을 교시하는 것이 아니라, 가상 환경 조립 시스템을 이용한 로봇 프로그래밍을 하는 데, 가상 환경 속에서의 부품 조립 시 발생하는 반발력을 사용자에게 인터페이스함으로써 작업의 효율을 증대시키는 데 목적이 있다.

2. 이전에 한 일

가상 환경 조립 시스템 개발을 계속적으로 진행해 왔었다. 여기서는, 기존의 로봇 중심의 언어가 아닌 물체 중심으로 작업을 기술할 수 있는 Task-level 제어언어 개발에 중점을 두었고, 또한 이를 효과적으로 로봇 제어언어로 번역하기 위한 Task-level 조립 계획 시스템의 구현방법을 제안하였다. 특히 각각의 완성품을 이루기 위한 로봇제어언어를 완성품 데이터 베이스로 구축하여 제안된 시스템이 실시간 특성을 가지도록 하였다. 또한 물체의 3차원 기하학적 특성의 데이터 베이스화, 로봇의 동역학을 고려한 최적의 경로계획 등의 연구가 수반되었다[3,5].

이 연구에서 개발된 가상 환경 조립 시스템은 3차원 컴퓨터 그래픽스 기술을 이용해서 개발되었는데, 시각 정보만을 이용하여 3차원 공간에서 부품을 조립하게 되므로 정확하게 조립하는 것이 아주 중요하게 되었다. 하지만 사용자에게 시각적인 정보만을 주고 3차원 공간상에서 정확하게 조립하길 바라는 것은 어려운 일이다. 따라서, 본 논문에서는 사용자에게 힘 되먹임을 전달함으로써 보다 효율적인 작업이 되도록 하는데 중점을 둔다.

본문 내용을 간략하게 소개하면, 3 장에서는 다중 접촉점을 표

현하기 위한 기본 접촉점을 대한 정의를 소개하고, 반발력을 어떻게 계산하는지 소개하겠다. 4 장에서는 3 차원 공간에서 어떻게 접촉이나 충돌이 일어났는지에 대한 이야기를 하며, 5 장에서는 시뮬레이션을 통해서 3 장과 4 장에서 이야기한 내용을 확인하겠다. 끝으로 결론과 앞으로 할 일에서 대해서 논하겠다.

3. 기본 접촉점에 대한 정의

일반적으로 실제 환경을 사실대로 표현해서 접촉점을 정의한다는 것은 아직은 불가능한 일이다. 따라서 본 논문에서는 가상 환경에 대해서 몇 가지 가정을 한다. 첫째, 사용자가 다루는 물체는 하나 뿐이다. 둘째, 장애물은 가상 환경에 고정되어 있다. 셋째, 물체들 사이에 정적인 반발력만을 고려한다. 넷째, 계산량을 줄이고 단순화 작업을 위해서 동적인 요소(friction, gravity, inertia ...)들은 생략한다. 다섯째, 다면체 모형 근사법에 의해서 물체를 모델링 한다.

위의 5 가지를 가정한다면, 접촉점의 종류를 크게 2 가지로 나눌 수 있게 된다.

3.1 평면-꼭지점 접촉

그림 1 은 한 물체의 평면에 다른 물체의 꼭지점이 접촉한 상태이다. 모든 물체의 표면은 수직한 방향으로만 작용하는 일정한 힘을 가지는 스프링이 덮고 있다고 가정하자. 또한 사용자에게 반발력을 주기 위해서는 크기와 방향이 필요하다. 크기는 물체 사이에 겹쳐지는 영역이 되고, 방향은 평면에 대해서 수직한 방향이 된다.

$$\mathbf{f} = -\mathbf{K} \mathbf{dx} \quad (1)$$

$$\mathbf{K} = k \mathbf{I} \quad (2)$$

\mathbf{K} 는 한 물체에 대해서는 같은 값을 가지므로, $1/k = 1/k_1 + 1/k_2$ 로 정의하게 되고, \mathbf{I} 는 3×3 단위 행렬이다. 가상 환경에서 평면-꼭지점 접촉이 발생하게 되면 겹쳐지는 영역을 찾게 되고, 관통 벡터 \mathbf{dx} 를 구하게 되면 반발력은 식 (1),(2)에 의해서 구해지게 된다.

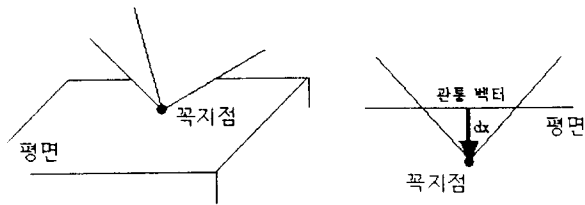


그림 1. 평면-꼭지점 접촉
Figure 1. Plane-Vertex Contact

3.2 모서리-모서리 접촉

그림 2 는 모서리-모서리 접촉을 설명하고 있다. 한 물체의 모서리가 다른 물체의 모서리와 서로 엇갈려 있으므로 접촉 상태를 이루고 있는 것을 보여주고 있다. 위의 절에서 설명한 것과 같은 방법으로 반발력이 만들어 진다.

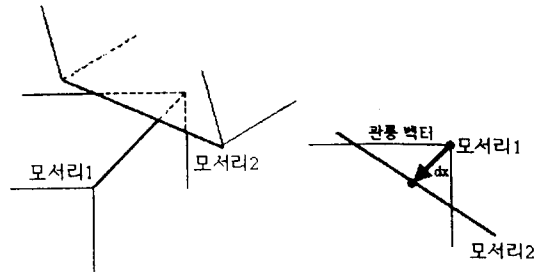


그림 2. 모서리-모서리 접촉
Figure 2. Edge-Edge Contact

3.3 다중 접촉에 대한 예

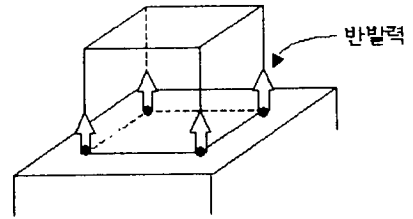


그림 3. 평면과 평면의 접촉(평면-꼭지점 4 개)
Figure 3. Plane-Plane Contact (four PV)

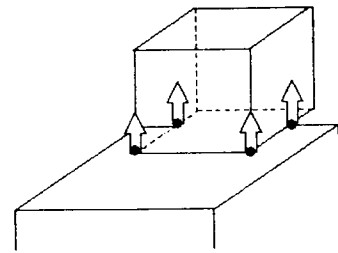


그림 4. 평면과 평면의 접촉 (평면-꼭지점 2 개, 모서리-모서리 2 개)
Figure 4. Plane-Plane Contact (two PV, two EE)

실제로 두 물체가 접촉했을 때 어떻게 정의된 접촉들에 의해서 표현되는지 보겠다. 그림 3 은 보이는 데로 4 개의 평면-꼭지점 접촉으로 표현되고, 그림 4 는 2 개의 평면-꼭지점과 2 개의 모서리-모서리 접촉으로 표현된다. 이와 같이 다면체 모형 근사법에 의해서 모델링되는 모든 물체들은 평면-꼭지점 접촉과 모서리-모서리 접촉인 2 개의 기본 접촉만으로 모든 접촉을 표현 가능하다.

3.4 반발력 계산

위에서 정의된 기본 접촉에 의해서 구해진 각각의 반발력들을 어떻게 합칠 것인가? 또 사용자에게 어떻게 회전력을 만들어 줄 것인가? 이에 대한 해답은 그림 5 에 있다. 정의된 기본 접촉에 의해서 만들어지는 반발력은 식 (3)에서 보는 것과 같이 단순히 방향에 맞게 합쳐 주지만 하면 된다. 하지만 회전력을 얻어내기 위해서는 사용자가 다루고 있는 물체의 중심에서 반발력이 발생하는 위치까지의 벡터를 구한다. 이 구한 벡터와 기본 접촉에 의해서 만들어진 반발력 벡터를 가지고 식 (4)와 같이 계산을

해주면 회전력을 얻어낼 수 있다.

$$\mathbf{f}_{total} = \sum_{i=1}^n \mathbf{f}_i \quad (3)$$

$$\boldsymbol{\tau}_{total} = \sum_{i=1}^n \mathbf{r}_i \times \mathbf{f}_i \quad (4)$$

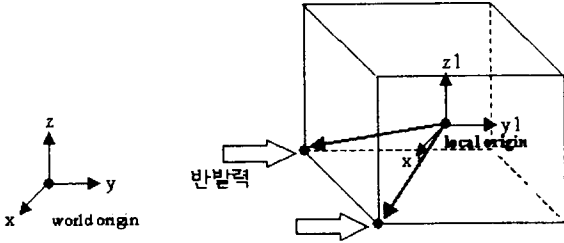


그림 5. 여러 개의 반발력을 합치기
Figure 5. Combine with reflection-forces

이와 같이 본 논문에서 정의된 기본 접촉 2 가지를 이용하게 되면 사용자에게 되돌려 보내야 되는 힘과 회전력을 구하는 데 별 어려움이 없게 된다. 하지만 그림 6 과 같은 상황이라면 좀 달라진다. 물체 1 과 물체 2 가 그림 6 과 같이 충돌 상태인데, 이전 상태를 알지 못한다면 경우 1 처럼 힘을 되돌려 보내야 되는지, 아니면 경우 2 처럼 만들어야 하는지 알 수 없기 때문이다. 따라서, 시뮬레이터는 사용자가 다루는 물체의 이전 상태를 알고 있어야만 한다.

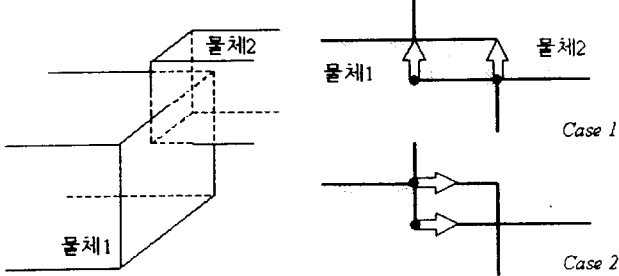


그림 6. 두 물체의 충돌
Figure 6. Collision with two object

4. 접촉 및 충돌 검사

가상 환경 시뮬레이터에서 계산량을 가장 많이 필요로 하는 것이 사용자가 다루는 물체와 장애물과의 충돌 여부를 검사하는 것이다. 본 논문에서는 계산량을 최대한 줄이기 위해서 2-모드 검색 방법을 제안했다. 사용자가 다루는 물체는 대부분 자유 공간을 움직이다가 장애물과 접촉 및 충돌을 하고 나서 결합하게 된다. 따라서 시뮬레이터가 움직이는 모든 시간동안 접촉 및 충돌점을 찾는 것이 아니라 일단 1 단계로 접촉 및 충돌 여부를 검사하고 2 단계로 접촉 및 충돌점을 찾는 것이다.

그림 7은 1 단계 검사 방법을 자세히 보여주고 있다. 3 차원 공간을 xy 평면, yz 평면, zx 평면으로 정사영을 시킨다. 정사영 시킨 평면에 장애물과 사용자가 다루는 물체 사이에 겹쳐지는 영역을 찾는 것이다. 그래서, 만약 3 평면 모두에서 겹쳐지는 영

역이 나타나면 3 차원 공간에서 접촉이나 충돌이 발생하였음을 알려주는 것이다. 이와 같이 1 단계가 만족되면 2 단계로 넘어가서 장애물과 다루는 물체에 대한 기하학적인 정보를 이용하여 접촉점이나 충돌점을 찾는 것이다[2,4].

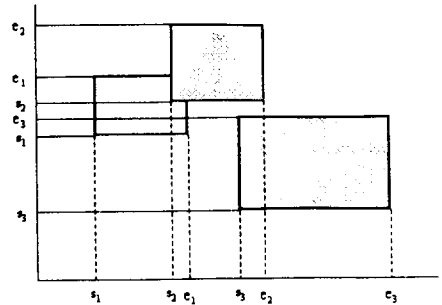


그림 7. 1 단계 충돌 검사 (계산량 감소)
Figure 7. Rough Check (reduce to computation)

그림 8은 반발력을 만들어 내기 위한 전체 알고리즘을 보여주고 있다. 1 단계 검사(Rough Interaction Check)를 통해서 계산량을 줄이고, 2 단계 검사에서는 접촉점이나 충돌점을 찾아서 정의된 기본 접촉에 따라 분류하고, 반발력을 계산해서 사용자에게 보내는 것이다.

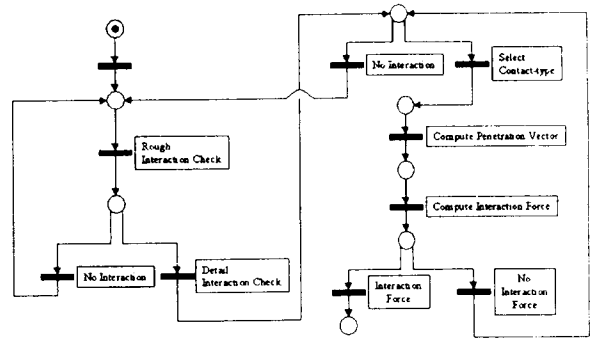


그림 8. 반발력에 대한 전체 알고리즘 (2-모드)
Figure 8. Total Algorithm on Feedback Force (two-mode)

5. 시뮬레이션

시뮬레이터의 기본 환경은 펜티엄 150MHz, 그래픽 라이브러리 OpenGL 을 기반으로 했다. 시뮬레이션을 간단하게 하기 위해서 물체는 육면체로 제한했다. 가상 환경하에서는 부품을 조립하게 되면 그 부품은 장애물로 인식하게 된다. 사용된 부품의 크기는 20x20x20, 30x30x30, 40x40x40 이고, 모든 부품의 표면을 둘러싸고 있는 스프링의 강도(stiffness)는 3,000 으로 했다.

첫번째 그림은 사용자가 다루고 있는 물체가 자유 공간에 있기 때문에 반발력이 발생하지 않으므로 막대 그래프가 보이지 않는다. 하지만 충돌이 일어나면 나머지 그림과 같이 반발력이 발생하게 된다.

시뮬레이터에서 화면 갱신은 부품의 변화가 생길 때마다 발생하며, 반발력을 계산하기 위한 2-모드 알고리즘은 20msec 마다 검색하게 된다.

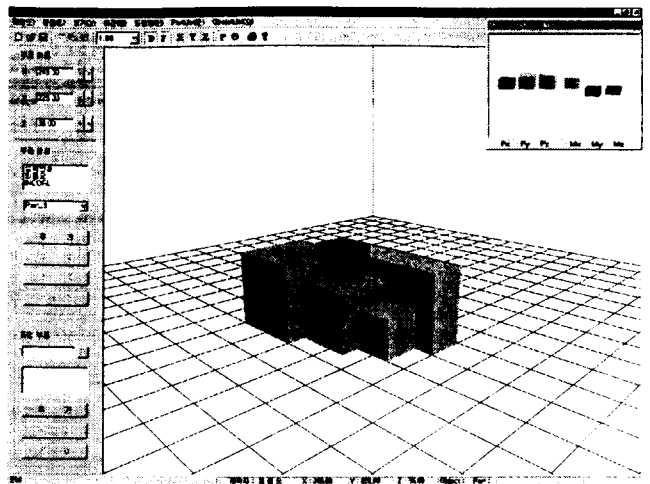
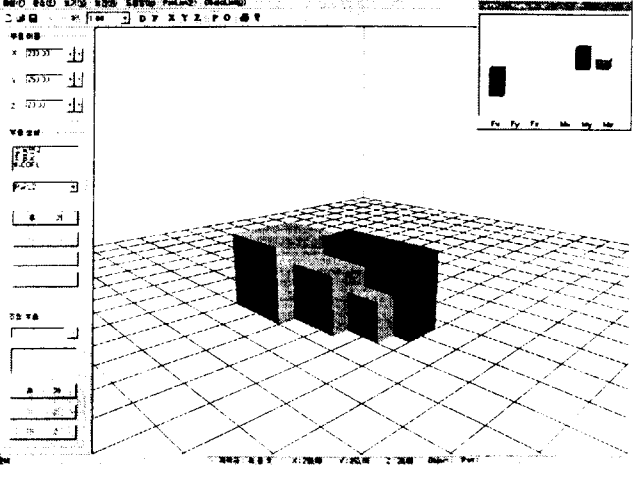
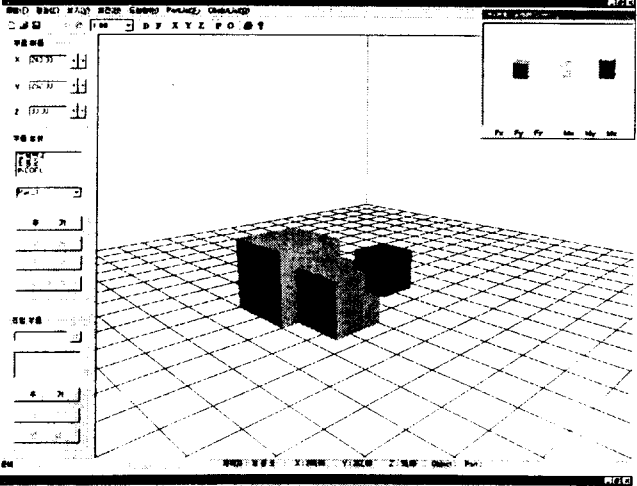
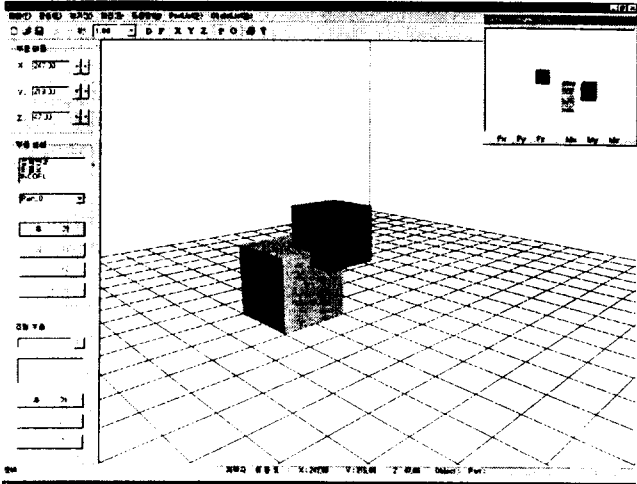
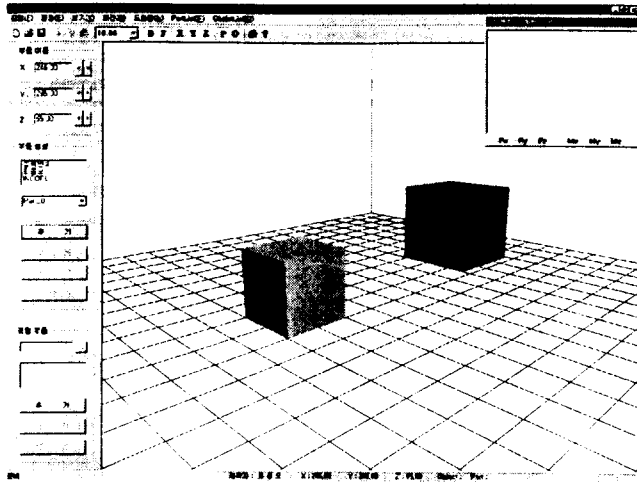


그림 9. 시뮬레이터

Figure 9. Simulator

6. 결론

본 논문에서 제안한 기본 접촉 2 가지를 이용하면 가상 환경에서 다면체 모형 근사법으로 표현되는 물체의 모든 접촉이나 충돌을 표현해 줄 수 있고, 반발력을 만들어 낼 수 있음을 시뮬레이터를 통해서 볼 수 있었다. 또한 제안된 2-모드 알고리즘을 이용하면 불필요하게 증가되는 계산량이 없어지므로, 더욱 빠른 응답을 제공하게 된다. 기존의 조립 시스템에 제안된 방법이 추가되면 사용자는 시각뿐만 아니라, 물체들 사이의 접촉이나 충돌에 대한 정보를 추가로 얻을 수 있게 되므로, 편리하고 정확할 뿐만 아니라 시뮬레이터에서의 조립 시간을 단축할 수 있으므로 전체 공정 시간이 줄어들게 된다.

참고문헌

- [1] G. C. Buradea, "Force and Touch Feedback for Virtual Reality," *A Wiley-Interscience Publication*, 1996
- [2] T. KOTOKU, K. TANIE, and A. FUJIKAWA, "Environment Modeling for the Interactive Display used in Telerobotic systems," *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1991.
- [3] H. W. Kim, B. H. Hwang, J. S. Park, I. H. Suh, and B. J. Yi, "Development of a Task-Level Assembly Planning System for Dual Arms," *Proc. of Intelligent Manufacturing Systems*, 1997
- [4] M. C. Lin, D. Mocha, J. Cohen, and S. Gottschalk, "Collision Detection: Algorithms and Applications," *Proc. of Algorithms for Robotics Motion and Manipulation*, 1996
- [5] 서일홍, 정재훈, 조상규, 김영욱, 황병훈, 김진오, "조립작업을 위한 로봇의 지능적 작업 수준 계획 시스템," *제어계측·자동화·로보틱스 연구회 합동 학술 발표회 논문집*, Vol. 4, No. 3, pp. 436-440, March 1995
- [6] 최한호, "Haptic 인터페이스 현황," *전자공학회지*, pp. 148-155, February, 1998.