

The Virtual Environment Control using Real-time Graphic Deformation Algorithm

姜 園 贊\* · 金 南 伍\*\* · 崔 昌 柱\*\*\*  
(Won-Chan Kang · Nam-Oh Kim · Chang-Ju Choi)

**Abstract** - In the established virtual-reality system, although it is possible to transact a faculty of sensation and graphic in a single PC, virtual object forcibly treated with rigid body for the reason of the huge amount of calculation, and the number of polygon is restricted. Furthermore, there is some difficulty in the financial aspect and a program field, because the existing virtual-reality system needs at least two workstations or super computers. In this study, the new force-reflecting algorithm called as "Proxy" and a finite element method of Hyperion are applied to this system in order to transact in real-time. Consequently, though the number of polygon, which brings about an obstacle is increased in the real-time graphic transaction, this system makes it possible to transact in the real-time, not being influenced by the size of the virtual object.

**Key Words** : Haptic, Graphic Deformation, FEM, Force-Display System, Virtual Reality

1. 서 론

인간이 받아들이는 외부의 수많은 정보들의 약 70[%] 정도가 시각을 통해 흡수되기 때문에 가상현실 연구에서는 시각화 시스템에 치우쳐 연구되었다. 하지만, 현실세계에서의 모든 작업들은 시각뿐만 아니라 인간이 손가락 끝에서 느끼는 역학적인 임피던스의 변화 즉, 접촉감각에 의해 이뤄진다. 따라서 이러한 감각정보는 인간이 어떤 복잡한 조작을 하는데 있어서 시간과 오차를 감소시켜준다. 인간이 가상공간에서 작업을 행할 때 사용자의 위치 및 힘 정보를 가상세계에 반영하고, 그에 따른 작업환경의 감각정보를 인간에게 제공하는 연구가 필요하게 되었다[1,2]. 가상현실에서 역감소소를 추가하게 되면 뇌에 의해서 처리되는 정보가 증가하고, 이런 정보의 증가는 특정 작업을 수행하는 데 걸리는 시간과 오차를 감소시킬 수 있음을 의미한다[3]. 따라서 가상현실에서 보다 실제에 가까운 현실감을 제공하여 작업자가 작업속도 향상 및 효율성을 높일 수 있는 시스템에 대한 연구가 필요하게 되었다.

기존의 연구에서는 단일 PC에서 역감 및 그래픽 처리가 가능한 시스템도 있지만, 계산량이 많아 가상물체를 강제로만 취급하였고, 가상물체의 폴리곤수도 제한이 있었다. 또한 가상물체의 실시간 변형을 위해서 2대 이상의 워크스테이션 또는 슈퍼컴퓨터에서 처리하는 형태를 취하고 있으므로, 장비의 구성 가격이나 프로그램상의 어려움 등으로 가상현실 연구의

제한이 있었다. 따라서 본 연구에서는 촉각 및 역감을 실시간으로 제시하기 위해 필요한 인터페이스 장치의 설계와 함께 새로운 힘처리 알고리즘인 프락시(Proxy) 알고리즘을 적용하여 안정된 힘처리를 수행하였다[4,5]. 가장 문제시되었던 실시간 그래픽 처리를 위해 계산량은 줄이고, 현실성은 향상시키도록 유한요소법을 도입한 시스템을 개발하였다. 유한요소법은 본래 시간이 많이 소요되는 방법이지만 하이퍼리온 방법을 도입하여 시간 소요문제를 해결하였다. 개발된 시스템은 USB 제어를 사용했는데 인간의 촉각 인식범위인 320[Hz]의 주파수 대역이 구현되었고, 상업화가 가능한 시스템이다.

실시간 처리가 가능하도록 설계된 6DOF 촉각장치와 USB 제어를 토대로 연구를 수행하여 이 시스템이 안정적으로 동작함을 확인하였고, 아울러 제안한 힘처리 알고리즘도 기존 연구에서 그 안정성이 증명되었다. 특히, 본 연구에서는 실시간 처리의 장애를 일으키는 그래픽처리에서 하이퍼리온 프로젝트에서 제안한 유한요소법을 이용해 수행하였고, 이 방식의 그래픽 변형알고리즘은 가상객체의 폴리곤수가 증가하더라도 작업영역으로 설정된 부분만을 가지고 처리함으로써 가상객체의 크기에 영향을 거의 받지 않고 처리가 가능하므로, 실시간 처리가 가능함을 시뮬레이션을 통해 증명하였다. 마지막으로, 제안된 시스템을 통한 가상물체 제작 실험에서 촉각이 지원되는 가상환경하에서 3차원 그래픽 모델을 제작하는 것이 시간상으로는 2배 이상의 제작시간을 단축하였다. 이는 눈에 보이지 않는 부분도 촉각이 보완해 줌으로 인해 보다 효과적으로 가상물체를 제작할 수 있다는 것을 의미한다. 차세대 O/S가 그래픽 위주로 발전한다고 할 때 3차원 인터페이스가 가능하고, 물체의 재질감 반향을 통한 도구의 조작 및 운용 등에서 제안된 시스템이 매우 효과적이라고 판단된다.

\* 正 會 員 : 朝鮮大 情報制御計測工學部 講師 · 工博

\*\* 正 會 員 : 朝鮮理工大 電氣科 教授 · 工博

\*\*\* 正 會 員 : 朝鮮大 電氣工學科 教授 · 工博

接受日字 : 2003年 3月 27日

最終完了 : 2004年 3月 5日

2. 촉각처리시스템

2.1 USB 제어기

가장 최근에 주목받고 있는 통신 방식으로는 IEEE 1394와 USB로, 이들 통신 방식을 통해 고속 데이터 처리가 가능한 시스템의 개발이 다두되고 있다. 물론 IEEE 1394 방식이 전송 속도가 400[Mbps]이고, USB v2.0 규격의 경우는 480[Mbps]의 전송 속도를 보이고 있다. 하지만 본 연구에서는 USB v1.1 벌크 모드로 제작하였다.

그림 1에서 엔코더 카운터부는 8비트 2채널을 갖는 LS 7266R1을 병렬로 연결하여 16비트 분해능을 얻고, D/A부는 12비트 4채널을 갖는 DAC7724를 사용하였다. 여기서 DC 모터의 정·역을 전환을 위해 D/A의 출력이 -10~0[V]는 역방향 회전, 0~10[V]는 정방향 회전을 하도록 설정을 하였다.

기존 연구에서는 PC의 ISA포트를 통해서 각 장치들을 직접 제어했지만, USB 방식이나 PCI 방식에서는 불가능하므로 USB 포트로 전송된 패킷을 처리하여 원하는 동작을 하도록 16비트 마이크로프로세서인 히다치사의 H8-3048을 사용하였다. 차후 USB 브리지 칩을 2.0 버전으로 업그레이드할 경우 1[kHz]의 처리가 가능할 것으로 판단된다. 그러나 현재 1.1 벌크 모드는 500[Hz]로 처리 가능하므로 재질감 표현에서 일반 사용자의 감각을 속이는 것은 가능하였다.

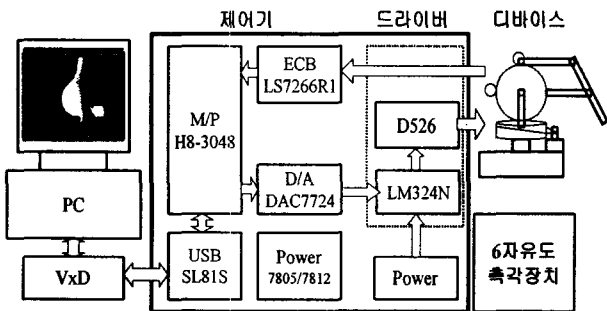


그림 1 USB 제어기의 블록 다이어그램  
Fig. 1 Block diagram of USB controller

2.2 개선된 힘처리 알고리즘

그래픽 렌더러는 단순하게 30[Hz]의 인터럽트 주기에 따라 메모리 상에 구현된 가상물체 프리미티브를 계산하고, 이를 모델링하여 3차원적으로 그려주는 특징을 가진다. 이는 실시간 처리로 인식하기에 충분한 주기를 가지고 있다. 그러나 점성을 가진 물체의 표현을 위해서는 변형된 유한요소법을 사용한다[6]. 햅틱 렌더러의 빠른 갱신에 대응하기 위해 그래픽 렌더링에 역적(力積)알고리즘을 추가하여 안정된 그래픽을 제공한다[7].

기존의 방식으로는 정적인 가상세계의 표현은 가능 하지만, 동적인 세계는 표현하지 못하는 문제를 가지고 있다[8]. 예로 가상세계에 물체를 띄우고 그것이 손가락과 충돌한다고 가정하면 손가락과 충돌한 후 물체의 움직임은 손가락 위치 측정 샘플링 시각이 각각 달라서 차이를 발생할 수 있다. 따라서 기존 방식에서는 샘플링 시각의 영향으로 안정하게 가

상세계를 갱신하지 않을 수도 있다. 그러므로 이의 개선을 위해 햅틱 렌더러와 그래픽 렌더러 사이에 역적제어기를 추가하여 해결하였다.

3. 실시간 그래픽 변형 알고리즘

그래픽 처리에서 강제 변형은 유한요소법에 기초를 두고 만들었다. 이 방법은 계산량이 많은 각종 파라미터 행렬과 그 역행렬을 미리 구해놓고, 실행시 강제와 노드의 이동에 적용된 스트래스 사이의 선형 관계를 찾는 것이다. 이 과정은 응용프로그램의 초기화 단계에서 처리되고, 이 선형관계는 행렬로 표현하여 파일로 저장된다[9,10]. 로딩시 파일에서 행렬을 추출해 분석하고, 실행시 충돌발생을 힘 벡터로 변환한다. 이 힘 벡터의 곱은 적재된 행렬이 응용프로그램에서 높은 정밀도로 강제의 변형을 찾는 것이 가능하다.

3.1 행렬연산의 간소화

하이패리온 프로젝트의 수행순서에서 행렬 연산은 초기화 시에 자동적으로 계산이 되는데, 사용자는 단지 가상물체(입방체)에 대한 기본 파라미터를 파일형태로 제공한다[11]. 1단계에서 각 유한요소의 강성 행렬이 발견된다. 2단계에서는 전역행렬 K를 얻는데 24x24를 갖는 8개의 행렬을 얻는다. 3단계에서는 전역행렬을 간소화한다. 가정에 따라 위의 행렬들이 묘사되고, 가상물체는 베이스에 고정된 것으로 가정한다. 이는 힘이 가상물체 위에서 적용하여도 결코 베이스는 움직이지 않도록 한다. 따라서 가상물체의 베이스 위의 9 노드를 나타내는 27개의 행렬 성분을 0으로 설정할 수 있다. 간소한 후에 57x57 크기의 경계 행렬 K'를 얻는다. 문제에서 주어지지 않는 것은 57x57 크기의 변위 벡터 U'에 의해 표시된다. 이 방법에서 가장 연산이 많은 단계인 4단계는 행렬 K'의 역행렬을 구하는 것이다. 행렬의 역변환이 얼마나 잘되었는가가 정확한 해답을 찾는지의 관건이 될 것이다. 5단계인 렌더마스킹 단순화(Render Mask Simplification)는 비디오 게임에서 사용하는 방법 중 하나로 행렬연산의 속도를 개선할 수 있다. 이는 가상물체에서 보이지 않는 부분은 그것의 이동을 아는 것이 소용없으므로 그래픽으로 보여줄 필요가 없다. 따라서 주어지지 않은 행렬 U'와 같은 크기를 가지는 단순화 행렬 RM이 계산된다. RM행렬 각 성분의 값은 0이나 1이다. 여기서 0은 관련된 노드의 기하학적인 부분을 숨기는 것을 의미하고, 1은 노드가 보여지는 것을 의미한다.

$$U = \begin{bmatrix} u'1 \\ u'2 \\ \vdots \\ u'9 \\ u'10 \\ u'11 \\ u'12 \\ u'13 \\ \vdots \\ u'57 \end{bmatrix} RM = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \rightarrow Ur' = \begin{bmatrix} u'1 \\ u'2 \\ \vdots \\ u'9 \\ 0 \\ 0 \\ 0 \\ u'13 \\ \vdots \\ u'57 \end{bmatrix} \rightarrow Ur = \begin{bmatrix} u'1 \\ u'2 \\ \vdots \\ u'9 \\ u'13 \\ \vdots \\ u'57 \end{bmatrix} = \begin{bmatrix} u'r1 \\ u'r2 \\ \vdots \\ u'r9(1) \\ u'r10 \\ \vdots \\ u'r57 \end{bmatrix}$$

결국, 우리는 51열과 54행을 가지는 행렬 K'-1r을 얻는다. 여기서 r은 인덱스로서 간소화한다. 새로운 행렬이 정방행렬이 되지 않고, 알려지지 않은 중심 노드를 제거할지라도 스트

레스가 가상물체의 중앙에 작용했을 때의 이동을 발견할 수 있다.

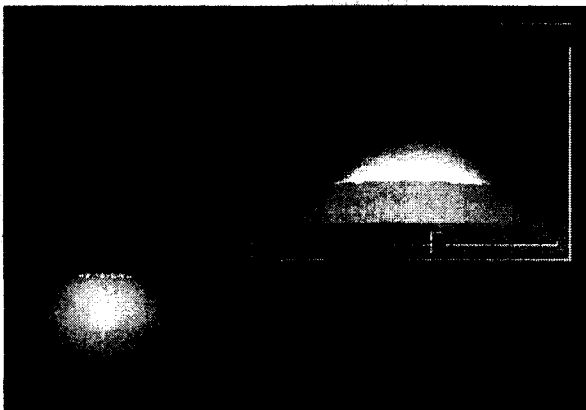
마지막 6단계는 프로그램의 초기화시 사용하도록 행렬 K'-1r을 파일로 저장하는 것이다. 이와 같은 전처리 과정을 통해 하이패리온 프로젝트에서는 가상물체를 얻어 K'-1r행렬로 저장하는 일을 서버에서 행하고, 클라이언트에서는 이를 호출하여 이용한다. 하지만 본 연구에서는 가상환경이 아닌 가상객체에 대한 변형을 하기 때문에 하나의 프로세서의 전처리 부분에서 이 과정을 수행하고, 후처리에서 변형을 수행한다.

### 3.2 유한요소법에 의한 영역선택 및 처리

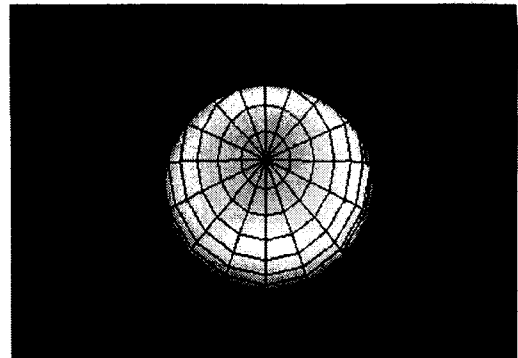
가상물체와 접촉을 가지는 가상점을 일정한 크기를 갖는 구로 나타내어 가상물체의 공간내에 있는 특정한 영역을 변형영역으로 선택하는 방법이다. 물론 이 크기는 조절할 수 있다. 이 공간영역 선택법에 의해 변형영역을 선택하여 프락시 알고리즘에 의해 힘을 제시하고, 그래픽 알고리즘에 의해 그래픽으로 제시하여 사용자의 거리감각 문제를 개선하였다.

그림 2의 (a)는 변형 영역을 선택할 때 미리 다른 각도에서 변형후의 모양을 제시하여 정확한 변형 영역을 선택하게 하고, (b)는 변형을 일으키는 구 모양의 가상점 크기를 조절하여 영역을 선택한다. (c)는 연속적으로 구를 변형시키고 있는 상태로 실시간으로 변형이 가능함을 나타낸다. (d)는 위의 과정을 되풀이하여 원하는 형태를 만들고 변형영역을 선택하는 구를 제거한 상태의 그림이다. 따라서 변형을 위해 이미 고정된 영역을 사용하는 대신 변형 구를 통해 변형 영역을 연속적으로 가변시키는 방법을 통해 실시간 그래픽 변형이 가능하게 하고 있다. 영역을 선택하는 순서와 방법은 다음과 같다.

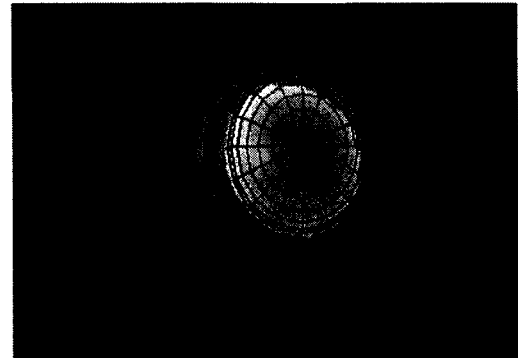
1. 프락시 알고리즘에서 가상물체의 표면상에서 움직이는 통상 가상점이라고 불리는 가상 이동점을 구한다.
2. 대상물에 부여되는 변형영역과 엔드포인트의 자세를 통해 변형된 가상물체를 가시화.
3. 2에서 변형된 영역내의 모든 정점을 1의 엔드포인트의 주변에서 시작해 순서적으로 검출해 변형영역으로 선택한다.



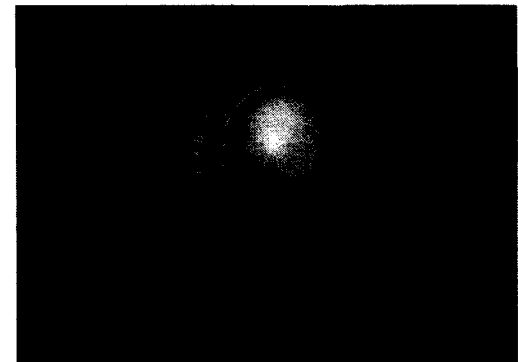
(a) 변형 영역의 변형 강도



(b) 변형 영역의 선택



(c) 연속적인 변형과 변형 영역선택



(d) 변형후의 가상물체

그림 2 변형 영역의 선택

Fig. 2 Selection of deformation area

### 3.3 실시간 그래픽 렌더링

그래픽 처리에 사용된 알고리즘의 처리 절차는 그림 3에 나타내었다. 하이패리온 프로젝트에서는 전처리와 후처리로 나누어서 처리를 해야 함으로써 사용자에게 매우 불편함을 야기 시켰다. 따라서 본 연구에서는 이 알고리즘을 개선해 전처리와 후처리를 포함해 전처리에서 작성된 가상물체에 대한 재질정보를 바탕으로 각종 행렬을 계산하여 저장하고, 후처리에서 다시 불러 표시하는 형식으로 진행되었다.

하이패리온 프로젝트는 그래픽만의 처리로 제한되었지만, 본 연구에서는 햅틱 렌더링이 주목적이고 그래픽은 이차적인 목적이었다. 그렇지만 실험 부분에서 제시하는 시뮬레이션에서처럼 제한된 폴리곤 내에서는 실시간 촉각 처리와 그래픽

변형을 이룰 수 있음을 확인 할 수 있었다. 다만 행렬의 역 변환시의 시간 지연으로 인해 발생하는 지연을 보상하기 위해 임시 파일로 작성하여 한번 사용한 가상물체는 그 재질행렬을 유지하고, 처음 사용한 행렬도 전처리 부분에서 파일로 저장하여 사용하였고, 후처리의 처리속도 향상을 위해 메모리에서 각종 행렬과 조건행렬들을 임시 저장하여 속도향상을 기하였다.

다만 변형 영역의 크기에 따라 속도차가 있으므로 전체적으로 조건을 적용하는 것은 앞서의 알고리즘에서 제시했듯이 실시간 처리에 제한을 주게된다. 따라서 본 연구에서처럼 촉각이 수반된 그래픽 변형에서는 제한적인 범위(변형영역으로 선택된 영역) 내에서 사용해야 하는 문제도 있다. 마찬가지로 역감의 제시에서도 실제와 같은 힘을 묘사하기 위해서는 그에 상응하는 출력을 낼 수 있는 촉각장치가 필요하나 본 촉각장치의 최대 힘인 10[Nm]로 제한하여 사용하였다.

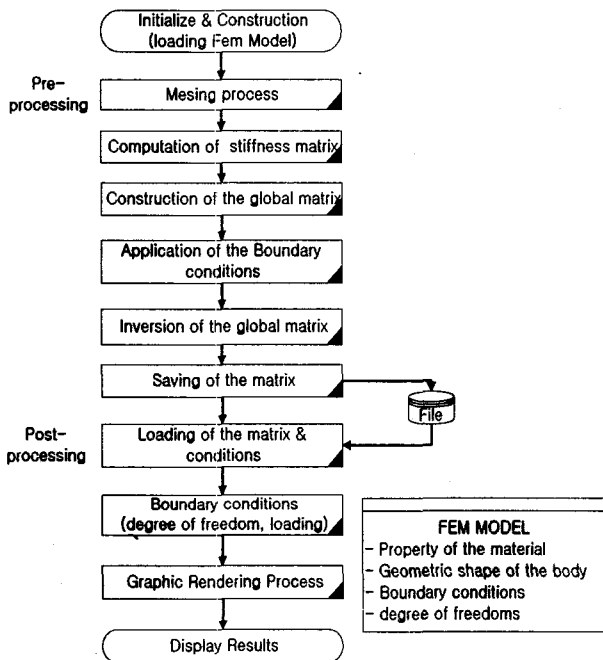


그림 3 실시간 그래픽 변형의 절차  
Fig. 3 A procedure of real-time graphic deformation

4. 실험 및 고찰

4.1 실험 시스템의 구성

본 연구에서 사용한 실험 시스템의 사진을 그림 4에 나타내었다. 사용자가 촉각장치의 엔드이펙터를 잡고 컴퓨터에서 생성된 가상의 물체를 시각적으로 인식하면서 가상물체와 상호작용하기 위해 가상환경 내에서 작용점을 움직인다. 이 때 로터리 엔코더에서 관찰각을 검출하고, 이 위치정보를 엔코더 카운터에서 읽어들이어 순가구학 해석에 의해 작용점의 위치좌표를 계산한다. 그리고 햅틱 렌더러에서 계산된 정보를 가지고 가상물체와의 접촉여부를 검출하게 되며, 충돌시 접촉방향에 따라 프락시 알고리즘을 사용해 x, y, z축 방향의 힘을 출

력한다. 만약 가상물체와 엔드이펙터가 충돌하게 되면 가상물체를 관통한 거리에 비례한 힘이 알고리즘들에 의해 값을 계산하여 드라이버를 통해 모터에 구동 토크를 제공하고, 다시 이를 30[Hz]마다 그래픽 변형 처리를 통해 가상세계를 보여주도록 하고 있다.

화면에 나타난 가상물체와의 충돌을 그래픽 렌더러에서 감시하다가 충돌이 감지되면 그래픽 변형과 함께 힘이 출력되는데, 이는 프락시 알고리즘에서 가상 프락시점을 통해 변위 변터를 계산하고, 이 변위 벡터를 통해 출력할 힘과 그래픽의 변형량이 결정되어 사용자는 실제와 같은 가상환경을 느낄 수 있게 된다.

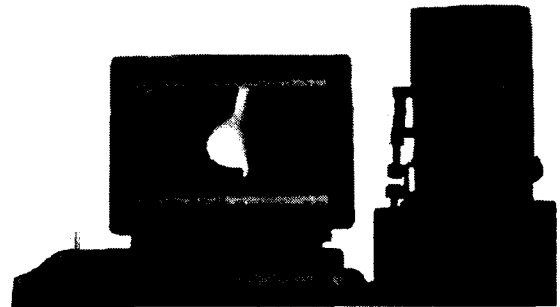


그림 4 실험 시스템의 구성  
Fig. 4 Composition of experiment system

4.2 알고리즘의 구성

실제 좌표계를 그래픽 좌표계로 변환하는 어파인(affine) 변환, 촉각장치의 엔드포인트, 가상물체와 가상물체간의 충돌을 감지하는 충돌검출 알고리즘, 가상세계의 현상을 그래픽으로 보여주는 그래픽 변형 알고리즘, 가상물체에서 촉각장치로 되돌려줄 힘을 계산하는 프락시 역감 렌더링 알고리즘, 모니터 상에 가상물체의 변형을 표현해주는 하이패리온 유한요소법을 이용한 그래픽 렌더링 알고리즘 등으로 구성하였다. 전체 알고리즘의 구성은 그림 5와 같으며 월드 매니저가 이들 알고리즘을 필요시마다 호출하여 처리한다. 다만 각종 파라미터 행렬 등과 위치정보 등을 공유 메모리 상에서 계속 교환해 가며 안정된 동작이 유지된다.

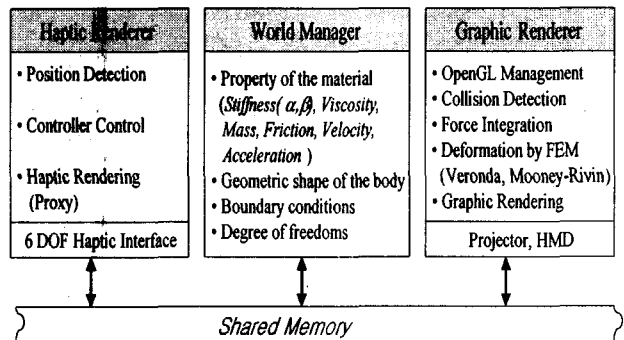


그림 5 알고리즘의 구조  
Fig. 5 The structure of algorithm

모든 동작은 엔드포인트 위치와 가상점으로 작용하는데 이들 위치가 같을 때는 충돌이 일어나지 않은 상황이므로 아무런 동작을 하지 않고 단순히 엔드 포인트의 위치만 변경하게 된다. 하지만 가상점의 위치가 엔드포인트와 일치하지 않을 경우는 충돌이 일어날 경우임으로 이 두점간의 벡터 크기만큼의 그래픽 변형과 힘의 충돌이 발생하도록 월드매니저가 총괄하여 수행하는 구조를 갖는다.

### 4.3 프로그램 구성

실험에 사용한 컴파일러는 마이크로소프트사의 Visual C++ enterprise v6.0으로 그래픽 라이브러리는 OpenGL (GLUT v3.7.3)을 사용하였다. 운영체제는 Windows 2000 advanced server (P-III 800E Dual, 768M 메모리)를 이용하여 작성되었다.

주요 클래스는 3D, Haptic, Hase, WorldMan, Application 등의 5개의 클래스로 구성되고, 각 클래스는 각각 쓰레드로 구동된다. 3D 클래스는 그래픽 라이브러리와 촉각장치의 좌표를 일치시켜 주는 좌표계산을 한다. 이는 그래픽좌표나 행렬연산의 계산 수행을 줄여 보다 최적화 하도록 배려했다. 컨트롤러 제어를 위한 Haptic 클래스는 VxD 가상장치 드라이버를 통해 USB 제어기를 통제하고, 프락시 알고리즘에 의해 위치 계산과 힘 출력을 담당한다. Hase 클래스에는 OpenGL 라이브러리를 통해 그래픽을 렌더링하고, 본 연구에서 개발된 그래픽 변형 알고리즘을 통해 실시간 변형을 처리하도록 하였다. WorldMan 클래스에서 이들 클래스의 구동시간을 각각 제어하여 안정성 있는 시뮬레이션이 가능하게 하고 있다.

본 연구에서 개발된 라이브러리는 입체 그래픽 표현, TCP/IP 네트워크 프로토콜을 사용하여 서버와 클라이언트로 나눠 분산처리 하는 고전적인 촉각처리 시스템을 구축할 수도 있지만, 현재의 PC 능력으로는 제안된 프로그램으로도 단일 PC에서 사용가능 함을 실험을 통해 증명하고 있다. 표 1에 클래스 구조를 정리하였다.

표 1 클래스 구조

Table 1 Class structure

3D	3D 좌표변환
Haptic	USB 컨트롤러 제어, Proxy 알고리즘
Hase	OpenGL 라이브러리, 그래픽 렌더링, 역적제어
Worldman	각 인터럽트별 클래스 관리
Application	변형실험 프로그램

### 4.4 그래픽 변형 처리 실험결과 및 고찰

그래픽 처리에 있어서 단순한 변형을 위한 일반적인 방법은 폴리곤수가 작으면 작을수록 효과적이거나 반대로 폴리곤수가 증가하면 할수록 지수적으로 증가함을 알 수 있다. 따라서 유한요소법을 이용한 경우도 전체적인 계산량 면에서 선형증가 함을 나타내고 있다. 하이패리온 프로젝트에서 제시한 것처럼 계산량을 대폭 줄이기 위해, 전처리 과정에서 필요

한 파라미터 행렬들을 구하고, 실행 상태에서 필요한 부분만의 계산을 수행하므로 폴리곤수의 영향을 대폭 줄일 수 있었다. 그림 6에서처럼 변형된 방법을 통해서 30,000개 이상의 폴리곤에서도 실시간 처리가 가능함을 나타내고 있지만 PC의 성능이 나날이 증가함에 따라 단일 프리미티브가 아닌 복수의 프리미티브들에 대해서도 실시간 처리가 가능할 것으로 기대된다.

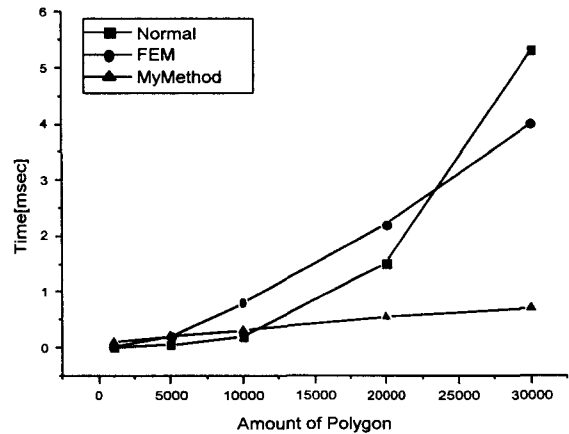


그림 6 그래픽 알고리즘의 결과  
Fig. 6 Result of graphic algorithm

### 4.5 촉각이 지원되는 가상환경실험

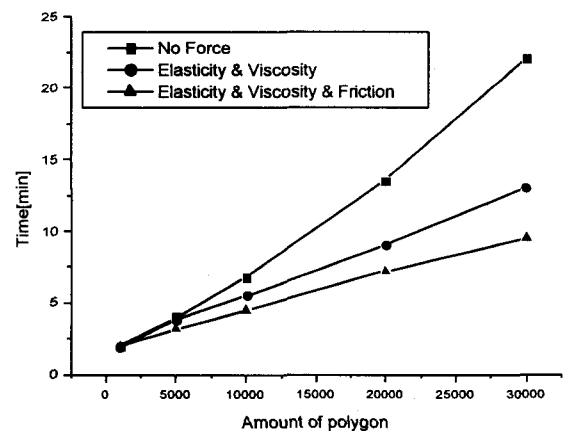


그림 7 작업 능력 특성  
Fig. 7 Characteristics of work ability

가상물체를 촉감이 부여된 상태와 부여되지 않는 상태에서 작업효과를 10명이 1주일간 테스트 프로그램으로 훈련한 후 힘 반영이 없는 경우, 점·탄성 반력만 있는 경우, 점·탄성외에 마찰력 등 현 연구에서 제시할 수 있는 가장 자연스러운 느낌을 가공 후, 가상물체에 부여하여 힘 반력이 있는 경우에 작업자가 얼마나 빨리 작업을 완료할 수 있는가를 테스트 한 결과를 그림 7에 나타내었다. 그림 7에서처럼 반력이 없는 경우에 비해 반력이 있을 경우 2~3배정도 빠른 작업 효과가 있음을 입증할 수 있었다.

5. 결 론

본 논문에서는 1대의 PC를 이용하여 안정된 역감 처리와 실시간 그래픽 변형을 동시에 처리하며 촉각의 인식범위를 만족시키는 시스템을 개발하였다. 이를 위해서 HIR 6DOF 촉각장치와 USB 방식의 제어기, 전류원에 기반을 둔 임피던스 컨트롤이 가능한 드라이버를 이용하였다. 역감 렌더링에는 빠른 충돌검출과 정확한 힘을 계산해내는 프락시(Proxy) 알고리즘을 적용하였고, 하이퍼리온 프로젝트에서 제시한 유한요소법 기반의 새로운 그래픽 변형 알고리즘으로 실시간 그래픽 처리를 구현하였다. 생성한 가상환경과 제작된 디바이스와의 인터페이스를 통해 시스템이 안정되게 동작함을 실험을 통해 증명하였다.

특히 실시간 그래픽 변형 프로그램을 이용한 가상물체 인터페이스 실험에서 증명된 것처럼 촉각이 지원되는 환경하에서 3차원 그래픽 모델을 제작하는 것이 시간상으로는 2배 이상의 제작시간을 단축하였고, 눈에 보이지 않는 부분도 촉각이 보완해 줌으로 인해 보다 효과적으로 가상물체를 제작할 수 있었다. 차세대 O/S가 그래픽 위주로 발전한다고 할 때 3차원 인터페이스가 가능하고, 물체의 재질감 반향을 통한 도구의 조작 및 가상현실 응용시스템 등에서 개발된 시스템이 매우 효과적이라고 판단된다.

향후 연구과제로는 단일 가상 객체를 대상으로 한 변형의 한계를 벗어나 가상환경 전체에 대해 처리 가능하도록 개발이 이뤄져야 한다. 또한 USB 버전 2.0 방식의 제어기의 도입과, 회전 및 병진 운동할 수 있는 시스템을 구축할 필요가 있다. 보다 높은 몰입감 배가를 위해 HMD와 VRD 등 새로운 시각장치들을 구동 가능한 라이브러리 제작을 통해 보다 사실적인 가상공간을 구축하고, 이를 이용해 다양한 시뮬레이터를 제작하는 데 있다.

감사의 글

이 논문은 2002년도 조선대학교 학술연구비 지원을 받아 연구되었음.

참 고 문 헌

[1] 강원찬, "실시간 촉각처리를 위한 프락시 및 그래픽 디포메이션 알고리즘," 朝鮮大學校 工學博士學位論文, 2003. 2.  
 [2] C. Ziles, "Haptic Rendering with the Tool handle Haptic Interface," Master's thesis. Massachusetts Institute of Technology, Dept. of Mechanical Eng., 1995, May  
 [3] D. C. Ruspini, et al., "The Haptic Display of Complex Graphical Environments," SIGGRAPH 97, Los Angeles, 1997.  
 [4] 강원찬, 김영동, "PC 기반의 6자유도 촉각장치의 개발," 대한전기학회 논문지, 제50D권, 제5호, pp.211-217, 2001.  
 [5] 강원찬, 신석두, 김영동, "실의 장력을 이용한 역감장치," 대한전기학회 논문지, 제50P권, 제4호, pp. 192-198, 2001.

[6] S. Fisher, M. C. Lin, "Fast Penetration for elastic bodies using deformed distance fields," IROS2001.  
 [7] T. Yoshikawa, T. Okamoto, "Display of Operating Feel of Virtual Tools in Rigid Frictional Contact with Environment," 日本計測自動制御學會論文誌 Vol. 32, No. 5, pp. 741-749, 1996.  
 [8] J. K. Salisbury, et al., "Constraint based God Object Method for haptic Display," Proc. of the IEEE/RSJ International Conference on Intelligent Robotics and System, 1995.  
 [9] M. B. Nielsen, S. Cotin, "Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation," Euro graphics '96, ISSN 1067- 7055, pp. 57-66, 1996.  
 [10] G. Picinbono, H. Delingette, N. Ayache, "Non-linear Anisotropic Elasticity for Real-Time Surgery Simulation," INRIA report de recherche No. 4028, October, 2000.  
 <http://www.inria.fr/rrrt/rr-4028.html>  
 [11] P. Rebours, "Real time deformation of solids," <http://membres.lycos.fr/lagrandeporte/hyperion>

저 자 소 개



강원찬 (姜 園 贊)

1970년 12월 23일생. 1993년 조선대학교 공대 제어계측공학과 졸업(학사). 2000년 동 대학원 제어계측공학과 졸업(석사). 2003년 동 대학원 졸업(박사)  
 Tel : 062) 230-7764  
 E-mail : kwc89@shinbiro.com



김남오 (金 南 伍)

1967년 2월 9일생. 1994년 조선대 공대 전기공학과 졸업. 1996년 조선대 대학원 전기공학과 졸업(석사). 1999년 동 대학원 전기공학과 졸업(공박). 현재 조선이공대학 전기과 조교수  
 Tel : 062) 230-8388  
 E-mail : nokim@mail.chosun-c.ac.kr



최창주 (崔 昌 柱)

1940년 7월 26일생. 1966년 조선대학교 전기공학과 졸업. 2000년 전남대학교 대학원 전기공학과 졸업(공박). 1993년 미국 Missouri 대학교 객원교수. 현재 조선대학교 전기공학과 교수.  
 Tel : 062) 230-7026  
 E-mail : cjchoi@chosun.ac.kr