

# 다양한 형태의 데이터를 포함하는 하이브리드 환경을 위한 안정적이고 사실적인 햅틱 제시 알고리즘

김종필<sup>1</sup>, 류제하<sup>2</sup>

광주과학기술원 기전공학과 인간-기계-컴퓨터 인터페이스 연구실<sup>1,2</sup>

{lowtar<sup>1</sup>,ryu<sup>2</sup>}@gist.ac.kr

## High-Fidelity Stable Haptic Interaction for Hybrid Virtual Environments

Jong-Phil Kim<sup>1</sup>, Jaha Ryu<sup>2</sup>

Human-Machine-Computer Interface Lab. GIST<sup>1,2</sup>

### 요약

본 논문은 다양한 형태의 객체데이터를 포함하는 하이브리드환경에 대한 안정적이고 사실적인 햅틱 제시 방법을 제안한다. 제안된 방법은 가상객체를 기술하는 방법에 의존하지 않고 일관된 방법으로 충돌검출 및 반력계산을 수행한다. 따라서 사용자 및 개발자는 부가적인 노력 없이 다양한 콘텐츠를 활용할 수 있으며, 빠르고 쉽게 가상환경을 구축할 수 있다. 또한 제안된 방법은 멀티 스레드로 구현된 안정화 연산을 수행하며, 이를 통해 느린 햅틱렌더링 속도를 가지는 환경에 대해서도 안정적이고 사실적인 역감을 제시한다. 따라서 제안된 방법은 다양한 응용분야에서 햅틱기술을 보다 쉽고 보다 효과적으로 적용할 수 있는 기회를 제공할 수 있다.

Keyword : Haptic, Stability, Virtual Reality

## 1. 서론

최근 반도체 기술의 발전과 더불어 고성능의 개인용 컴퓨터들이 널리 보급됨에 따라 가상현실 및 이의 응용기술에 관하여 많은 관심이 기울어지고 있다. 가상현실은 컴퓨터상에 가상으로 생성된 환경을 사용자의 오감을 통해 경험할 수 있도록 해주는 기술을 의미한다 [1]. 초기의 가상현실 기술은 컴퓨터 그래픽스를 이용한 시각 정보 및 다 채널 청각 정보의 제시를 통해 현실감 있는 환경을 재현하는데 중점을 두었다. 이러한 컴퓨터 영상 및 대화형 인터페이스의 이용은 설계, 시뮬레이션, 그리고 생산 자동화 분야뿐만 아니라 의학 및 교육 분야 등에서 광범위하게 사용되고 있다. 그러나 단일방향의 정보전달을 의미하는 시각 및 청각 제시기술은 사용자와 가상환경과의 상호작용 시 제한된 기능만을 제공한다는 한계를 지니고 있다. 이후 양방향 정보전달을 포함하는 햅틱 기술

의 발전을 통해 사용자는 능동적으로 가상환경에 참여할 수 있게 되었다.

만지다 라는 의미의 그리스어 ‘haptesthai’로부터 유래된 햅틱(haptic)은 사용자에게 가상 혹은 실제 환경의 다양한 정보를 역감(kinesthetic) 및 촉감(tactile)을 통해 제공해주는 일련의 과정을 의미한다. 입력 및 출력장치인 햅틱인터페이스를 통해 제시되는 역/촉감 정보는 다양한 환경을 더 자세히 인지하도록 도와주며, 사용자의 몰입감을 증가시켜 효과적으로 환경내의 객체들을 느끼고 조작할 수 있도록 도와준다. 이러한 이유로, 다양한 분야에서 안정적이고 효율적인 햅틱 제시 알고리즘에 관한 연구가 활발히 이루어지고 있다.

다양한 햅틱 제시 알고리즘 가운데 형태기반 햅틱렌더링 알고리즘(geometry dependent haptic rendering algorithm [2])은 객체의 기하학적 정보를 역감을 통해 제공하여 사용자가 가상의 객체를 실제로 만들 수 있게 해 주는 알고리즘으로 충돌검

출 및 반력계산 등의 과정을 포함한다. 이것은 가상 객체를 기술하는 방법에 따라 곡면(surface) 기반 방법[3-11]과 볼륨(volume) 기반 방법[12-15]으로 분류된다. 전자는 폴리곤, 삼각메쉬, NURBS, 또는 음함수(implicit) 등으로 기술된 곡면 데이터를 위한 방법이며, 후자는 복셀(voxel) 등으로 기술된 볼륨 데이터를 위한 방법이다. 초기의 햅틱 렌더링 알고리즘은 HIP(haptic interaction point)와 가상객체 사이의 penetration depth 를 이용하여 반력을 계산하는 penalty method 로 시작되었다. 그러나 이 방법은 급격한 반력변화 및 얇은 객체에 대해 적절한 반력을 만들어 줄 수 없다는 문제점들을 포함한다. 이것을 보완하기 위해 곡면의 구속 조건을 따라 움직이는 IHIP(ideal haptic interaction point)를 이용하여 반력을 계산하는 constraint method[4]가 제안되었다. 대부분의 이러한 연구는 그래픽스 분야에서 흔히 사용되는 삼각메쉬와의 충돌검출 및 반력계산을 그 대상으로 하였다. 이후 음함수곡면[9,10] 및 복셀데이터[15]를 가지는 가상 객체와의 햅틱렌더링 알고리즘들이 제안되었다.

그러나 지난 10 여 년간의 활발한 연구결과에도 불구하고 햅틱은 많은 응용분야에서 시각 및 청각 제시기술에 비해 적극적으로 활용되지 못하고 있는 것이 사실이다. 이것은 저가의 범용 햅틱 디바이스의 개발 및 보급의 한계라는 문제뿐 아니라 다음과 같은 기존의 햅틱 제시 알고리즘이 가지는 특성에 기인한다. 첫 번째는 기존의 알고리즘들이 가상객체를 기술하는 방법에 따라 충돌검출 및 반력계산 방법을 달리한다는 점이다. 서로 다른 방법으로 기술된 가상객체에 대해 햅틱을 적용하기 위해서는 triangulation 또는 voxelization 등과 같은 데이터 변환을 통한 통합과정을 거쳐야 한다. 이것은 가상환경에 포함시킬 수 있는 콘텐츠의 제약을 불러오며, 많은 기존의 콘텐츠를 그대로 활용할 수 없음을 의미한다. 기존의 알고리즘이 지니는 이러한 면은 가상환경 구축에 많은 시간을 필요로 하게 만들며, 특히 실시간으로 갱신되는 객체데이터를 포함하는 환경에 대한 적용을 어렵게 만든다.

두 번째는 안정적인 역감제시를 위해 상당히 빠른 햅틱연산을 요구한다는 점이다. 햅틱 제시

알고리즘은 시각 및 청각 제시 알고리즘과 달리 양방향 에너지전달을 포함하기 때문에 항상 안정성 문제를 염두에 두어야 한다. 일반적으로 적절한 범위의 강성을 가지는 객체를 사용자에게 안정적으로 제시하기 위해서는 1KHz 이상의 햅틱렌더링 속도를 가져야 하는 것으로 알려져 있다 [1]. 빠른 햅틱연산은 가상환경에 포함되는 객체의 단순화를 통해 디테일한 정보가 생략된 역감을 제공함으로써 얻어지게 된다 [2]. 이것은 기존의 연구가 사용자에게 사실적인 힘을 제시하는데 어느 정도의 한계를 지니고 있음을 의미한다. 더욱이 의료용 시뮬레이터와 같이 동적 객체나 변형 가능한 객체를 포함할 경우 동역학 및 유한요소 해석 등과 같은 추가적인 연산을 수행해야 하며, 따라서 사용자에게 제공될 수 있는 사실감은 더욱 제한될 수 밖에 없다.

본 논문은 다양한 형태의 객체데이터를 포함하는 하이브리드환경에 대한 안정적인 사실적인 햅틱 제시 방법을 제안한다. 제안된 방법은 가상객체를 기술하는 방법에 의존하지 않고 일관된 방법으로 충돌검출 및 반력계산을 수행한다. 따라서 사용자 및 개발자는 부가적인 노력 없이 다양한 콘텐츠를 그대로 활용할 수 있으며, 빠르고 쉽게 가상환경을 구축할 수 있다. 또한 제안된 방법은 멀티 스레드로 구현된 안정화 연산을 수행하며, 이를 통해 느린 햅틱렌더링 속도를 가지는 환경에 대해서도 안정적인 사실적인 역감을 제시한다. 따라서 제안된 방법은 다양한 응용분야에서 햅틱 기술을 보다 쉽고 보다 효과적으로 적용할 수 있는 기회를 제공한다. 본 논문은 다음과 같이 구성되어 있다. 2 장에서는 제안된 방법의 전체 개요를 자세히 소개하며, 햅틱렌더링 알고리즘 및 안정화 알고리즘은 각각 3, 4 장에 기술하였다. 5 장에서는 제안된 방법을 직접 적용한 예와 그 결과를 설명하였으며, 마지막으로 6 장에는 결론과 향후 연구 과제를 기술하였다.

## 2. 알고리즘 개요

일반적으로 햅틱 시스템은 햅틱디바이스, 제어알고리즘, 그리고 가상환경으로 이루어진다. 가상환경은 주어진 데이터에 대해 역/촉감정보를 제공하기 위한 햅틱 렌더링 알고리즘과 시각정보를

제공하기 위한 그래픽 렌더링 알고리즘을 포함한다. 그리고 필요에 따라 동역학 및 유한요소 해석 등을 담당하는 시뮬레이션 엔진을 포함하기도 한다. 제어알고리즘은 햅틱디바이스를 구동하기 위한 함수들을 포함하며, 일반적으로 액츄에이터와 엔드이펙터의 관계를 결정하는 기구학 연산 등이 이에 해당된다.

제안된 햅틱 제시 방법은 그림 1에 나타난 것과 같이 그래픽 렌더링, 햅틱 렌더링, 그리고 제어 알고리즘을 담당하는 3개의 멀티 스레드로 구현된다. 주어진 객체 데이터들은 오브젝트버퍼 또는 추가적인 그래픽 렌더링을 통해 픽셀버퍼에 저장된다. 픽셀버퍼는 시각적으로 렌더링 될 수 있는 모든 객체에 사용될 수 있으며, 부가적인 노력 없이 이미 개발된 그래픽 프로그램 소스를 그대로 사용하여 사용자에게 역감을 제시해 줄 수 있는 편의성을 제공한다. 오브젝트버퍼는 삼각메쉬, 음함수곡면, 또는 복셀 데이터 등과 같이 모델링과정 및 선행계산을 통해 이미 잘 마련된 객체에 대해 사용된다.

충돌검출 및 반력계산은 InteriorQuery 함수를 이용하여 수행된다. InteriorQuery 함수는 오브젝트버퍼 및 픽셀버퍼를 참조하여 주어진 포인트가 객체 내부에 있는지를 검사한다. 햅틱 렌더링 알고리즘은 컨트롤버퍼로부터 현재의 HIP의 위치를 읽어오고 충돌검출을 수행한다. 컨트롤버퍼는 각각 독립적인 스레드로 실행되는 제어알고리즘과 햅틱렌더링 알고리즘간의 데이터 교환을 위해 사용된다. 충돌이 검출되면 LOMI (local occupancy map instance)를 생성하고 이것을 참조하여 IHIP의 위치를 계산한다. LOMI는 매우 작은 크기의 임시

적인 occupancy map으로 InteriorQuery 함수를 통해 갱신된다. 계산된 IHIP는 HIP에 가장 가까운 곡면상의 포인트를 의미하며, HIP와 IHIP의 관계를 통해 사용자에게 제시될 반력을 계산한다. 계산된 반력은 컨트롤버퍼에 저장되며 이러한 과정은 매  $T_h$  시간마다 반복된다.

제어알고리즘은 A/D를 통해 햅틱디바이스의 엔코더 값을 읽고, 순기구학 해석을 통해 엔드이펙터의 위치를 의미하는 HIP의 값을 계산한다. 계산된 HIP는 컨트롤버퍼에 저장된다. 안정화 알고리즘은 컨트롤버퍼에 저장된 반력 정보를 참조하여 사용자에게 안정적으로 제시될 수 있는 힘을 계산한다. 계산된 힘은 자코비안 연산을 통해 액츄에이터에서 발생시켜 주어야 할 커맨드 토크로 변환되어 햅틱디바이스에 전달된다. 이러한 과정은 매  $T_c$  시간마다 반복되며, 사용자는 가상환경 및 햅틱렌더링 속도인  $T_h$ 에 관계없이 항상 안정적인 역감을 제시 받을 수 있다.

### 3. 햅틱 렌더링 알고리즘

#### 3.1 픽셀버퍼를 위한 InteriorQuery

제안된 알고리즘은 가상객체를 기술하는 방법에 의존하지 않고 일관된 방법으로 충돌검출 및 반력계산을 수행한다. 단지 가상객체를 기술하는 방법에 따라 달라지는 것은 InteriorQuery 함수이다. InteriorQuery 함수는 오브젝트버퍼 및 픽셀버퍼를 참조하여 주어진 포인트가 객체의 내부에 있는지 외부에 있는지를 판별하는 함수이다. 삼각메쉬, 음함수곡면, 또는 복셀 데이터 등과 같이 모델링과정 및 선행계산을 통해 이미 잘 마련된 객체의 경우 오브젝트버퍼에 저장되어 햅틱연산에 사용될 수 있다. 이러한 객체에 대해 주어진 포인트가 객체 내부에 있는지를 판별하는 과정은 이미 기존의 많은 충돌검출 알고리즘에서 다루어왔다. 따라서 본 논문에서는 픽셀버퍼를 참조하여 InteriorQuery 함수를 수행하는 과정만을 자세히 설명하고자 한다.

픽셀버퍼는 이미 잘 마련된 객체뿐 아니라 시각적으로 렌더링 될 수 있는 모든 객체에 사용될 수 있다. 이것은 부가적인 노력 없이 이미 개발된 그래픽 프로그램 소스를 그대로 사용하여 햅틱연산을 수행할 수도 있음을 의미한다. 픽셀버퍼에

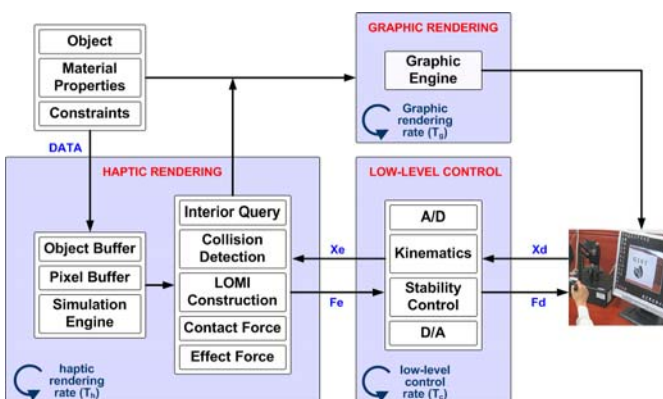


그림1. 제안된 햅틱시스템의 전체구성

할당된 객체는 추가적인 그래픽 렌더링을 통해 픽셀버퍼에 저장되며, 이 과정은 다음과 같은 두 가지 방법 중 하나로 수행된다. 첫 번째는 햅틱 렌더링이 수행되기 전 딱 한번 객체의 전체 영역을 저장하는 방법이며, 두 번째는 매 그래픽 렌더링마다 객체의 국부 영역을 저장하는 방법이다. 전자는 정적 및 동적 객체에 사용될 수 있으며, 후자는 정적, 동적, 변형 가능한 객체 및 실시간으로 갱신되는 데이터에도 사용될 수 있다. 이러한 방법으로 픽셀버퍼는 N 면의 depth 이미지를 포함하게 된다. 일반적으로 전체영역을 대상으로 할 경우 해당되는 객체의 바운딩박스 6 면에 위치한 가상카메라를 통해 6 면의 depth 이미지를 저장하며, 국부영역을 대상으로 할 경우 전체 햅틱 워크스페이스를 포함하는 바운딩박스의 6 면에 위치한 가상카메라를 통해 이전의 IHIP 주변의 depth 이미지를 저장한다.

픽셀버퍼를 위한 InteriorQuery 함수는 주어진 포인트가 N 면의 depth 이미지 내부에 있는지를 순차적으로 검사한다. 만일 주어진 포인트가 모든 depth 이미지에 대해 내부에 위치하면 true 를 리턴한다. 이것은 각각의 가상카메라를 통해 보았을 때 주어진 포인트가 객체의 내부에 위치함을 의미한다. 이 과정을 자세히 설명하면 다음과 같다. 먼저 각각의 depth 이미지에 대해 주어진 포인트에 해당하는 픽셀을 찾는다 (그림 2(a)). 픽셀 좌표계 및 오브젝트 좌표간의 관계는 사용된 가상카메라의 파라미터에 의해 결정된다. 다음으로 픽셀에 저장된 depth 값을 오브젝트 좌표계로 변환한다 (그림 2(c)). 일반적으로 픽셀에 저장되는 depth 값은 0 과 1 사이의 값을 가지며, 직교사영(orthogonal projection)의 경우 근절단평면(near clipping plane)부터 원절단평면(far clipping plane)까지 선형적으로 맵핑된다. 투시사영(perspective projection)의 경우 다음과 같은 관계로 맵핑된다.

$$Z = c - \frac{Z_{far} Z_{near}}{Z_{far} - d(Z_{far} - Z_{near})} \quad (1)$$

마지막으로 주어진 포인트와 변환된 depth 값을 비교하여 이것이 depth 이미지 내부에 있는지를 판별한다. (그림 2(b)) 이러한 과정을 통해 시각적으로 렌더링 되는 객체에 대해 쉽게 InteriorQuery 를 수행할 수 있다.

### 3.2 충돌 검출

형태기반 햅틱렌더링 알고리즘은 HIP 와 객체 간의 충돌 시 발생하는 반력을 계산하는 과정으로 충돌검출은 필수적이다. 앞에서 오브젝트버퍼 및 픽셀버퍼를 참조하여 주어진 포인트가 객체의 내부에 있는지를 조사하는 방법을 살펴보았다. 이것을 이용하면 HIP 가 오브젝트버퍼 및 픽셀버퍼에 저장된 객체의 내부에 있는지를 쉽게 판별할 수 있다. 그러나 HIP 하나를 이용하는 이러한 방법은 얇은 객체에 대해 HIP 가 매우 짧은 시간 (single haptic iteration time) 동안 뚫고 지나가 버릴 경우 충돌을 검출할 수 없다는 단점을 지닌다. 이러한 문제는 이전과 현재의 HIP 를 잇는 HIP 경로의 라인선분(line segment)과 객체간의 충돌을 검사하여 해결할 수 있다. 일반적으로 HIP 와 객체간의 충돌은 정지된 객체에 대해 HIP 가 움직일 때와 정지된 HIP 에 대해 객체가 움직일 때의 두 가지 상황의 조합으로 발생된다. 여기서는 이 두 가지 상황에 대해 충돌을 검출하고 temporal IHIP 를 위치시키는 방법에 대해 살펴본다. 다음에 설명될 LOMI 는 이렇게 구해진 temporal IHIP 를 중심으로 생성되며, 이 LOMI 를 참조하여 현재의 IHIP 를 결정한다.

#### CASE I. HIP 가 움직여 충돌이 발생된 경우

시간이 t일 때 객체와 HIP<sub>t</sub>간의 충돌검출 과정은 다음과 같다. 먼저 이전 t-1 에서의 IHIP<sub>t-1</sub>와 현재 t에서의 HIP<sub>t</sub>를 잇는 라인선분을 생성한다(그림 3(a)). 만일 이전시간 t-1 에서 충돌이 발생하지 않았을 경우 IHIP<sub>t-1</sub>는 HIP<sub>t-1</sub>와 일치한다. 다음으로

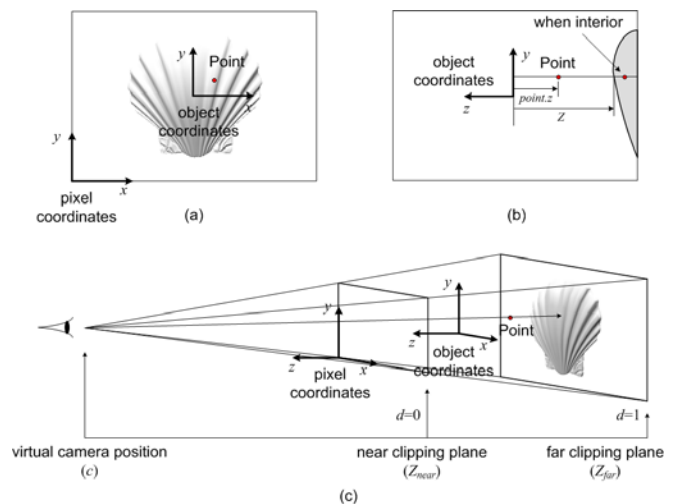


그림 2. 픽셀좌표계 및 오브젝트좌표계간의 관계

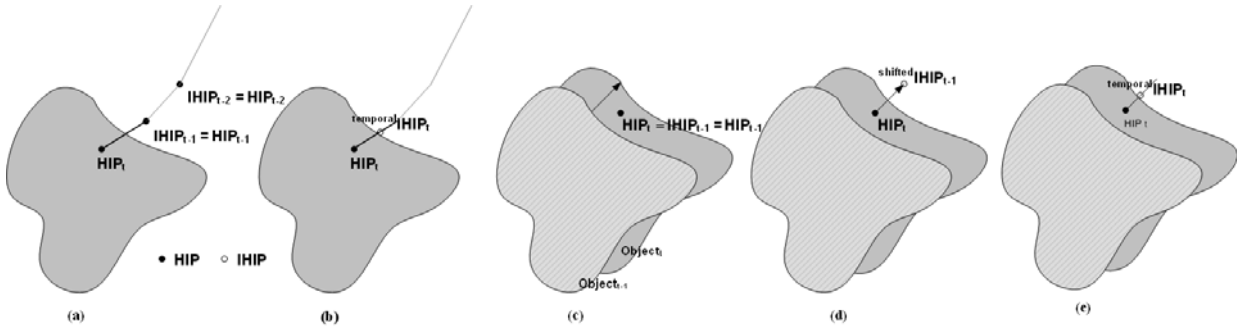


그림 3. 제안된 햅틱제시 방법에서의 충돌검출

생성된 라인선분상에 일정 간격을 가지는 포인트를 생성시킨다. 이것은 라인선분에 대한 복셀화된 표현을 의미하며, 본 논문에서는 LOMI내 복셀크기의 간격을 사용하였다. 마지막으로  $IHIP_{i,1}$ 로부터  $HIP_{i,1}$ 까지 생성된 포인트들에 대해 순차적으로 InteriorQuery를 검사하여 객체내부에 존재하는 포인트가 발견되면 충돌검출을 선언하고 최초의 내부포인트에 temporal IHIP를 위치시킨다(그림 3(b)).

### CASE II. 객체가 움직여 충돌이 발생된 경우

앞에서 살펴본 것과 반대로 정지된 HIP에 대해 객체가 움직여 충돌이 발생할 경우  $HIP_i$ 와  $IHIP_{i,1}$ 가 일치하기 때문에 HIP의 경로에 대한 라인선분을 생성시킬 수 없다(그림 3(c)). 따라서 얇은 객체가 매우 짧은 시간 (single haptic iteration time) 동안 HIP를 지나갈 경우 충돌을 검출할 수 없다. 이것을 보완하기 위해  $IHIP_{i,1}$ 대신 shifted  $IHIP_{i,1}$ 를 사용하여 라인선분을 생성시킨다. 여기서 shifted  $IHIP_{i,1}$ 는 객체의 운동을 참조하여  $IHIP_{i,1}$ 를 강제 운동(rigid body motion)시킨 포인트를 의미한다(그림 3(d)). 이것은 고정된 HIP에 대해 객체가 움직여 충돌이 발생한 상황은 고정된 객체에 대해 HIP가 정 반대로 움직여 충돌이 발생한 상황과 일치함을 의미한다. 이후 충돌검출 및 temporal IHIP를 위치시키는 방법은 앞의 경우와 동일하다(그림 3(e)).

### 3.3 반력 계산

제안된 알고리즘은 충돌검출 이후 반력계산을 위해 LOMI를 사용한다. LOMI는 매우 작은 크기의 임시적인 occupancy map으로 다양한 형태로 표현된 객체를 포함하는 하이브리드 환경의 햅틱렌더링에 필요한 통합된 데이터를 제공한다. 이것은 햅틱렌더링에 필요한 최소한의 크기로 생성되며 실시간으로 InteriorQuery 함수를 통해 갱신된다.

본 논문에서는 외부복셀 및 내부복셀에 대해 각각 0 과 1 이 할당된 1-bit 복셀모델을 사용하며 LOMI 좌표계를 구성하는 원점과 단위벡터들이 오브젝트 좌표계로 기술된다(그림 4). LOMI는 IHIP를 HIP에 가장 가까운 외부복셀에 위치시키기 위해 사용되기 때문에 LOMI의 크기는 single haptic iteration 시간 동안 IHIP가 움직일 수 있는 모든 범위를 포함해야 하며, 따라서 햅틱렌더링 속도와 사용자가 햅틱인터페이스를 조작하는 최대 속도를 고려하여 다음과 같이 설정한다.

$$LOMI \text{ size} \geq \frac{2 \times \text{maximum device speed}}{\text{haptic rendering rate}} \quad (2)$$

만일 LOMI가 식(2)에 기술된 것보다 작아지면 사용자는 최대속도에서 불필요한 drag force를 느끼게 된다. 예를 들어 사용자의 최대속도가 1m/sec라 가정하면 1KHz의 햅틱렌더링 속도에 대해 적어도 2mm의 크기를 가지는 LOMI를 사용해야 한다. 만일 0.5mm 크기의 복셀을 사용한다면 LOMI는 총 4x4x4개의 복셀을 포함하게 된다. 복셀의 크기는 사용되는 컴퓨터의 연산능력을 고려하여 결정한다.

이렇게 생성된 LOMI를 참조하여 반력계산을 수행하는 과정은 다음과 같다. 먼저 HIP와 객체간의 충돌이 검출되면 이때 결정된 temporal IHIP를 중심으로 LOMI를 위치시키고(그림 5(a)) 각

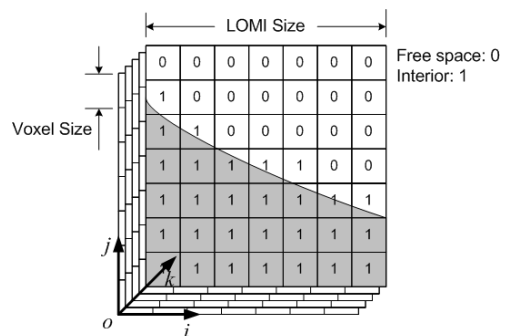


그림 4. LOMI (Local Occupancy Map Instance)

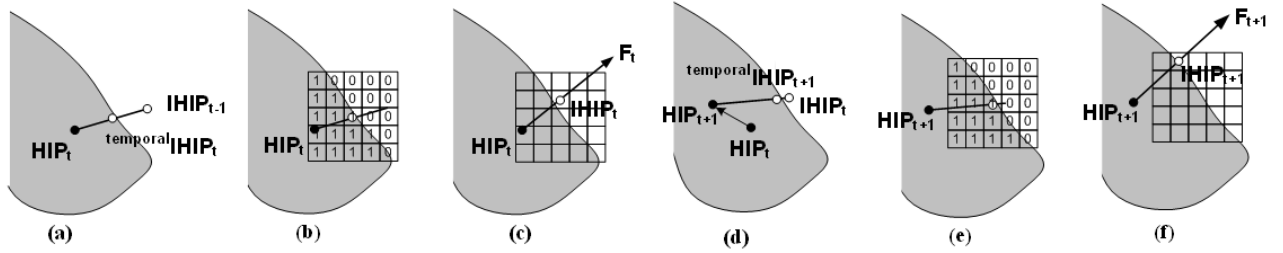


그림 5. 반력연산 수행 과정

복셀에 해당하는 포인트에 대해 InteriorQuery 를 수행하여 LOMI 를 갱신한다(그림 5(b)). 이 후 IHIP 는 LOMI 를 참조하여 HIP 에 가장 가까운 외부복셀에 위치시킨다(그림 5(c)). 사용자에게 제시 될 반력은 객체의 재료특성을 참조하여 HIP 와 IHIP 의 위치를 통해 계산된다. 이러한 과정은 매 햅틱 렌더링마다 반복된다(그림 5(d~f)). 계산된 반력은 컨트롤버퍼에 저장되어 컨트롤알고리즘으로 전달되며, 동적 또는 변형 가능한 객체의 경우 시뮬레이션엔진으로 전달되어 객체의 위치 또는 모양 등이 수정된다.

#### 4. 제어 알고리즘

햅틱 시스템은 양방향 에너지전달을 포함하기 때문에 이것의 안정성을 보장하는 것은 매우 중요한 문제이다. 또한 사용자에게 제시되는 역/촉감이 더욱 의미 있기 위해서는 높은 질의 역감을 계산할 수 있어야 한다. 이것은 햅틱 렌더링이 더 많은 연산을 수행해야 함을 의미한다. 특히 의료용 시뮬레이터와 같이 동적 객체나 변형 가능한 객체를 포함할 경우 동역학 및 유한요소 해석 등과 같은 추가적인 연산을 수행해야 하며, 따라서 햅틱 렌더링 속도는 상당히 떨어지게 된다. 그러나 햅틱 렌더링 속도가 떨어지게 되면 실제 사용자에게 안정적으로 제시될 수 있는 역/촉감의 임피던스 범위는 상당히 줄어들게 된다. 이것은 비록 햅틱 렌더링 알고리즘이 사실적인 힘을 계산해 낸다 하더라도 안정성 문제로 인해 결국 사용자는 제한된 범위의 역감을 느낄 수 밖에 없음을 의미한다.

이러한 한계를 극복하기 위해 제안된 햅틱 제시 시스템은 멀티 스테드로 구현된 새로운 안정화 알고리즘을 사용하며, 이를 통해 느린 햅틱 렌더링 속도를 가지는 환경에 대해서도 안정적이고 넓은 범위의 역감을 제시할 수 있다. 그림 6 은 멀티스테드로 구현된 안정화 알고리즘을 도식적으로 보여준다. 이 경우 전체 시스템의 안정성 조건은 다음과 같다 [16][17].

$$\sum_{k=0}^{n-1} b v_d^2(k+1) + \sum_{k=0}^{n-1} F_d(k) v_d(k+1) \geq 0 \quad (3)$$

여기서  $b$  는 햅틱디바이스에 존재하는 댐핑을 의미한다. 식(3)은 가상환경 및 샘플앤홀드에서 발생하는 에너지가 햅틱디바이스에서 전부 소모될 수 있어야 함을 의미한다. 이 조건을 만족시키기 위해 제안된 방법은 다음과 같은 제어알고리즘을 사용한다.

$$\begin{aligned} F_d(n) &= F_d(n-1) + \beta(n) v_d(n) \\ v_e(n) &= v_d(n) \end{aligned} \quad (4)$$

$$\text{where } \beta(n) = \frac{F_e(n) - F_d(n-1)}{v_d(n)} \text{ for } v_d(n) \neq 0$$

제어알고리즘은 컨트롤 버퍼로부터 반력( $F_e(n)$ ) 정보를 읽어와  $\beta(n)$ 을 계산하고, 식(4)를 통해 햅틱 디바이스에서 발생시켜 주어야 할 힘( $F_d(n)$ )을 결정한다. 이 경우 식(3)에 기술된 안정성 조건은 다음과 같이 표현될 수 있다.

$$\sum_{k=0}^{n-1} (b - \beta(k+1)) v_d^2(k+1) + \sum_{k=0}^{n-1} F_d(k+1) v_d(k+1) \geq 0 \quad (5)$$

이것은 안정성 조건 때문에  $\beta(n)$ 의 값이 제한되어

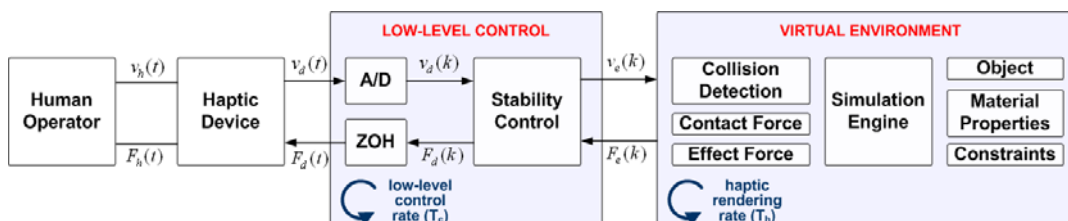


그림 6. 멀티 스테드를 이용한 안정화 알고리즘의 구현

야 함을 의미하며, 따라서  $F_d(n)$ 는 햅틱랜더링 알고리즘에서 계산된  $F_e(n)$ 와 다른 값을 가지게 된다. 본 논문에서는 햅틱 시스템의 안정성을 만족시키기 위해 다음과 같은 바운딩규칙을 사용하여  $\beta(n)$ 을 제한시킨다.

$$\begin{aligned} \text{if } \beta(n) > \beta_{\max}(n) \text{ then } \beta(k) &= \beta_{\max}(n) \\ \text{if } \beta(n) < \beta_{\min}(n) \text{ then } \beta(n) &= \beta_{\min}(n) \end{aligned} \quad (6)$$

식(6)과 같이 제한된  $\beta(n)$ 을 사용함으로써 안정성은 보장시킬 수 있으나 사용자는 햅틱랜더링 알고리즘을 통해 계산된 반력정보가 아닌 제한된 역감을 제시 받게 되고, 따라서 사용자에게 제공될 수 있는 사실감은 어느 정도 제약을 받게 된다. 그러나 햅틱 랜더링 속도( $T_h$ )보다 제어알고리즘 속도( $T_c$ )를 충분히 높여주게 되면  $F_e$ 에 비해  $F_d$ 는 훨씬 빈번히 갱신되며 결국  $F_d$ 는  $F_e$ 와 같은 값을 가지게 된다. 예를 들어 100Hz로 동작하는 햅틱 랜더링 알고리즘을 통해 10N만큼의 반력이 계산되었다고 가정하면,  $F_d$ 는 0N 이고  $v_d$ 는 1m/sec인 햅틱디바이스의 경우  $\beta(n)$ 은 10의 값을 가지게 된다. 그러나 안정성 문제로 인해 바운딩 규칙에 의해  $\beta(n)$ 이 2로 제한되었다면 사용자는 2의 힘만을 제시 받게 된다. 그러나 제어알고리즘의 속도를 5KHz로 높이면 사용자는 1msec(5번의 iteration)만에 10의 힘을 제시 받게 된다. 따라서 사용자는 햅틱랜더링 속도에 관계없이 이것에 의해 계산된 역감을 제시 받을 수 있다. 식(4)에 나타난 제어알고리즘은 매우 적은 연산을 요구하므로 충분히 빠른 갱신속도를 가질 수 있다.

식(5)에 기술된 안정성 조건을 만족시키기 위한  $\beta_{\max}(n)$  및  $\beta_{\min}(n)$ 은 다음과 같이 결정된다. 만일  $b \geq \beta(n)$ 을 만족하면 식(5)의 첫 번째 항은 항상 비음수의 값을 갖는다. 또한 다음의 관계를 만족하면 식(5)의 두 번째 항 역시 항상 비음수의 값을 갖는다.

$$F_d(n)v_d(n) \geq \frac{1}{2b}(F_d(n)+F_d(n-1))(F_d(n)-F_d(n-1)) \quad (7)$$

따라서  $b \geq \beta(n)$  및 식(7)의 관계를 만족시키는  $\beta(n)$ 을 선택함으로써 햅틱 시스템의 안정성을 보장시킬 수 있다. 식(7)에 나타난 조건은 식(4)를 통해 다음과 같이 기술 될 수 있다.

$$(\beta(n) - \gamma_{\max}(n))(\beta(n) - \gamma_{\min}(n)) \leq 0 \quad (8)$$

$$\begin{aligned} \text{where } \gamma_{\max}(n) &= b - \left( \frac{F_d(n-1)}{v_d(n)} \right) + \sqrt{b^2 + \left( \frac{F_d(n-1)}{v_d(n)} \right)^2} \\ \gamma_{\min}(n) &= b - \left( \frac{F_d(n-1)}{v_d(n)} \right) - \sqrt{b^2 + \left( \frac{F_d(n-1)}{v_d(n)} \right)^2} \end{aligned}$$

여기서  $\gamma_{\max}$ 는 항상  $b$ 보다 크거나 같으며  $\gamma_{\min}$ 는 항상  $b$ 보다 작거나 같다. 따라서 다음과 같은  $\beta_{\max}(n)$  및  $\beta_{\min}(n)$ 을 사용하여  $\beta(n)$ 을 제한시키면 사용자는 항상 안정적이고 사실적인 역감을 제시 받을 수 있다.

$$\begin{aligned} \beta_{\max}(n) &= b \\ \beta_{\min}(n) &= \gamma_{\min}(n) \end{aligned} \quad (9)$$

제안된 제어알고리즘의 성능을 살펴보기 위해 Impulse Engine™ [18]을 사용하여 가상벽 실험을 수행하였다. 가상벽 실험은 사용자에게 안정적으로 제시될 수 있는 힘의 임피던스 범위를 살펴보기 위해 일반적으로 사용되는 벤치마킹 테스트이다. 사용된 Impulse Engine™e의 경우 대략 10Nsec/m 정도의 댐핑을 가지고 있으며, 따라서 1KHz의 햅틱 랜더링에 대해 대략 20KN/m까지의 임피던스 범위를 안정적으로 제시할 수 있다. 그림 7의 (a)(b)(c)는 50Hz의 햅틱랜더링에 대해 제안된 제어알고리즘을 사용하지 않고 가상벽의 강성이 각각 500N/m, 700N/m, 그리고 1000N/m일 때 햅틱디바이스의 응답을 보여준다. 느린 햅틱 랜더링 속도에 의해 사용자가 안정적으로 제시 받을 수 있는 임피던스 범위는 700N/m이하로 현저히 줄어들게 된다. 그림 7의 (d)(e)(f)는 50Hz의 햅틱 랜더링에 대해 제안된 제어알고리즘을 사용하였을 때 햅틱디바이스의 위치, 반력, 그리고 사용자에게 실제로 제시되는 가상벽의 강성을 각각 보여준다.

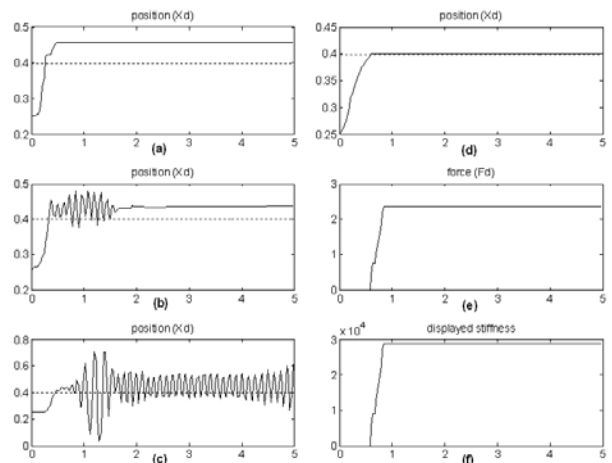


그림 7. Impulse Engine을 사용한 가상벽 실험

여기서 제안된 제어알고리즘은 2.5KHz의 속도로 구동되었으며, 이 결과를 통해 50Hz의 느린 햅틱 렌더링 속도에도 불구하고 사용자는 30KN/m이내의 임피던스를 안정적으로 제시 받을 수 있음을 확인할 수 있다. 이 결과는 제안된 알고리즘이 햅틱 렌더링속도에 관계없이 사용자에게 항상 안정적이고 사실적인 역감을 제시 할 수 있음을 의미한다.

### 5. 실제 적용의 예

이제까지 우리는 다양한 형태의 객체데이터를 포함하는 하이브리드환경을 위한 안정적이고 사실적인 햅틱 제시 방법에 대해 살펴보았다. 그리고 간단한 가상벽 테스트를 통해 제안된 제어알고리즘이 사용자에게 햅틱렌더링 속도에 관계없이 안정적이고 사실적인 힘을 제시할 수 있음을 확인하였다. 여기에서는 제안된 햅틱 제시 방법을 실제 가상환경에 적용한 결과에 대해 살펴보고자 한다. 이를 위해 6 자유도 PHANToM™ [19] 햅틱 디바이스 및 Pentium IV 3.2GHz가 사용되었다.

그림 8은 삼각메쉬 및 Z-cam™ [20]을 통해 얻은 depth 영상을 포함하는 하이브리드 환경에 대해 제안된 햅틱 제시방법을 적용한 예를 보여준다. 여기서 삼각메쉬로 구성된 TV모델은 픽셀버퍼에 할당하였으며 Z-cam™ 을 통해 얻은 depth 영상(그림 8(d))은 오브젝트버퍼에 저장하여 햅틱렌더링을 수행하였다. 이를 통해 사용자는 데이터 변환을 통한 통합 과정 및 계층적 데이터구조의 구성 등의 추가적인 노력 없이 모든 객체를 만지고 느낄 수 있다. 그림 8(a)는 사용자가 TV모델을 만질 때의 결과를, 그림 8(b)는 depth 영상을 만질 때의 결과를 각각 보여준다. 여기서 하늘색 구는 IHIP를 의미하며, 붉은색 선은 사용자에게 제시되는 힘의 크기와 방향을 나타낸다.

그림 9는 의료용 볼륨데이터에 대해 제안된 햅틱 제시 방법을 적용한 예를 보여준다. 여기서 볼륨데이터는 오브젝트 버퍼에 저장되었으며, Iso-Value 를 참조하여 수행되는 별도의 InteriorQuery 함수를 등록하여 햅틱렌더링을 수행하였다. 사용자는 볼륨데이터 내부를 이동하며 intensity 에 관련된 힘 정보를 제공받을 수 있으며(그림 9(a)) 이것과 더불어 IsoValue 에 해당하는 표면곡면에 해



(a)



(b)



(c)

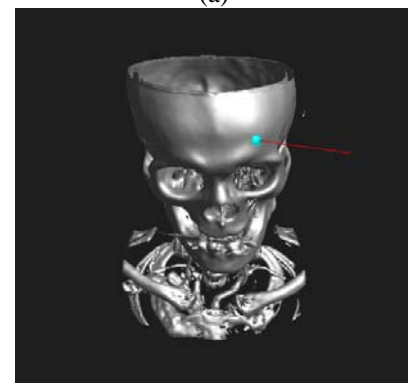


(d)

그림 8. 삼각메쉬 및 depth 영상에 대한 햅틱제시



(a)



(b)

그림 9. 의료용 볼륨데이터에 대한 햅틱 제시



당하는 역감을 제공받을 수 있다(그림 9(b)). 여기서 그림 9(a)와 (b)는 각각 의료용 볼륨데이터에 대해 volume rendering 및 ray-casting 의 방법을 사용하여 그래픽랜더링을 수행한 결과를 보여준다.

이러한 실제 적용의 예에서 햅틱 연산을 위해 필요로 되는 시간은 폴리곤의 개수, 볼륨 데이터 및 영상의 크기와 상관 없이 LOMI 를 구성하는 복셀의 개수에 의존하며, 0.2mm 크기의 복셀을 5x5x5 개를 포함하는 LOMI 를 사용하였을 경우 대략 0.07msec 의 시간을 필요로 함을 알 수 있었다. 이것은 제안된 방법이 거대한 가상환경에 대해서도 쉽게 사용될 수 있음을 의미한다. 좀 더 세밀한 크기의 복셀을 사용하였을 경우 사용자는 좀더 세밀한 역감을 제시 받을 수 있으며, 그만큼 더 많은 햅틱 연산을 요구한다. 그러나 느린 햅틱 연산으로 인해 안정적으로 제시될 수 있는 역감의 저하는 제안된 제어알고리즘을 통해 막을 수 있다.

## 6. 결 론

우리는 본 논문을 통해 다양한 형태의 객체 데이터를 포함하는 하이브리드환경을 위한 안정적이고 사실적인 햅틱제시 방법을 제안하였다. 제안된 햅틱 제시 방법은 그래픽 랜더링, 햅틱 랜더링, 그리고 제어 알고리즘을 담당하는 3 개의 멀티 스테드로 구현된다. 오브젝트버퍼 및 픽셀버퍼를 이용한 햅틱랜더링은 가상객체를 기술하는 방법에 의존하지 않고 일관된 방법으로 충돌검출 및 반력계산을 수행한다. 멀티 스테드로 구현된 제어알고리즘은 안정화 연산을 수행하며, 이를 통해 가상환경 및 햅틱 랜더링 속도에 관계없이 안정적이고 사실적인 역감을 제시한다. 이를 통해 사용자 및 개발자는 부가적인 노력 없이 다양한 콘텐츠를 그대로 활용할 수 있으며, 빠르고 쉽게 가상환경을 구축할 수 있다. 또한 느린 햅틱랜더링 속도를 가지는 환경에 대해서도 안정적이고 사실적인 역감을 제시할 수 있다. 따라서 제안된 알고리즘은 다양한 응용분야에서 햅틱기술을 보다 쉽고 보다 효과적으로 적용할 수 있는 기회를 제공할 수 있다.

## 감사의 글

This work was supported by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting IT Research Center (RBRC) and Smart Haptic Interface Project at Gwangju Institute of Science and Technology (GIST).

## 참고 자료

- [1] G. C. Burdea and P. Coiffet, Virtual Reality Technology, 2<sup>nd</sup> Edi, Wiley Interscience, 2003
- [2] K. Salisbury, F. Barbagli and F. Conti, "Haptic Rendering: Introductory Concepts", IEEE Computer Graphics and Applications Magazine, Special issue on Haptic Rendering, April 2004
- [3] T. Massie and K. Salisbury, "The PHANTOM Haptic Interface :A Device for Probing Virtual Objects," Proc. ASME Winter Ann Symp. Haptic Interfaces. Virtual Environment and Teleoperator Systems, vol. 55, no.1, pp. 295-300, 1994.
- [4] C. Zilles, K. Salisbury, "A constraint-based god-object method for haptic display", IEE/RSJ Int. Conf. Proc. Intelligent Robotics and System, vol 3, pp. 146-151, 1995.
- [5] D. C. Ruspini, K. Kolarov, and O. Khatib "Robust Haptic Display of Graphical Environments", Proc. ACM SIGGRAPH, pp.345-352, 1997.
- [6] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic Rendering: Programming Touch Interaction with Virtual Objects", Proc. Symp. Interactive 3D Graphics, pp. 123-130, 1995.
- [7] S. Rusinkiewicz and M. Levoy. "QSplat: A Multiresolution Point Rendering System for Large Meshes", Proc. ACM SIGGRAPH, pp. 343-352, 2000.
- [8] D.C. Ruspini, K. Kolarov, and O. Khatib., "The Haptic Display of Complex Graphical Environment", Conf. Proc. ACM SIGGRAPH, vol 1, pp. 295-301, 1997.
- [9] K. Salisbury and C. Tarr, "Haptic Rendering of Surfaces Defined by Implicit Functions", Proc. ASME. Dynamic Systems and Control Division, vol. 61, pp. 61-67, 1997.
- [10] L. Kim, A. Kyrikou, G.S. Sukhatme, and M. Desbrun, "An Implicit-Based Haptic Rendering Technique", IEEE/RSJ Int. Conf. Proc. Intelligent Robots and Systems, volume 3, pp. 2943-2948, 2002.
- [11] S. Walker and K. Salisbury "Large Haptic Topographic maps: Marsview and the Proxy Graph Algorithm", Proc. ACM Symp. Interactive 3D graphics, pp83-92, 2003.
- [12] R.S. Avila and L.M. Sobierajski, "A Haptic Interaction Method for Volume Visualization" J. IEEE Visualization, pp. 197-204, 1996.
- [13] D.A. Lawrence, C.D. Lee and L.Y. Pao, "Shock and Votex visualization Using a Combined Visual/Haptic Interface", J. IEEE Visualization, 2000.
- [14] H. Iwata and H. Noma, "Volume Haptization ," Proc. IEEE Symp. Research Frontiers in Virtual Reality, pp. 16-23, 1993.
- [15] W. McNeely, K. Puterbaugh, and J. Troy, "Six Degree-of-Freedom Haptic Rendering using Voxel Sampling", Proc. ACM SIGGRAPH, pages 401-408, 1999.
- [16] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi, "A Novel Theory for Sample Data System Passivity", IROS 2002, Lausanne, Switzerland, 2002
- [17] J. P. Kim and Jeha Ryu, "Energy Bounding Algorithm Based-On Passivity Theorem for Stable Haptic Interaction Control," 12th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, March, 2004
- [18] [http://www.immersion.com/industrial/joystick/impulse\\_engine](http://www.immersion.com/industrial/joystick/impulse_engine)
- [19] <http://www.sensable.com>
- [20] <http://www.3dvsystems.com>