

# 가상환경을 위한 힘피드백 제시기의 모듈 인터페이스 개발

김종국(고려대)\*, 박창훈(KIST IMRC),  
송재복(고려대), 고희동(KIST)

## Development of Module Interface of Force Feedback Controller for Virtual Reality

Jong-Kuk Kim \*(Korea Univ.), Chang-Hun Park(KIST IMRC)  
Jae-Bok Song(Korea Univ.), Hee-Dong Ko(KIST)

### 요 약

가상현실 기술은 이제 컴퓨터가 생성한 환경과 사용자간의 자유로운 상호작용을 수행할 수 있도록 하는 것이 중요하게 되었다. 이러한 상호작용을 위해서는 시각 제시기뿐만 아니라 사용자가 가상환경에서 물리적인 힘을 느낄 수 있도록 하는 것이 중요하다. 본 논문은 가상환경에서 사용자에게 운동감을 제시할 수 있는 기기들의 소프트웨어적인 아키텍처를 제안하고, 또한 이러한 기기들을 연결하는 데 있어서 가상환경과 통합될 수 있도록 가상환경 서버와 힘피드백 제시기간의 통신규약을 제시한다.

### Keyword

Virtual Reality, Human Factor, Device Server, Human Interaction Device, Force Feedback Controller, External Module Interface

## 1. 서 론

가상현실이란 컴퓨터와 인간 사이의 상호작용을 통하여 사용자로 하여금 실제감을 느끼도록 하는 기술이다. 이와 같은 기술은 교육, 오락, 방송, 의료, 훈련 등과 같은 다양한 분야에서 적용되고 있다. 가상현실 공간이 제공하는 실제감을 증대시키기 위해서는 기존의 시청각 위주의 가상환경에서 벗어나, 좀더 다양한 감각을 제시할 수 있는 외부 모듈이 필요

하다. 이러한 감각 제시 모듈을 이용한 시뮬레이터는 이미 국내에서도 여러 연구기관을 통해 개발되고 있다.<sup>(1,2,3,4)</sup>

그러나 이러한 모듈 제시의 문제점은, 가상현실 시스템과 모듈간의 성능 격차로 인한 성능 저하와 모듈간의 확장, 동기화로 인한 시간 지연, 각 모듈과 시스템 사이의 불안정성 등의 문제점이 발생한다.

가상환경과 모듈간의 성능 차이를 해결하기 위한 방법으로는 가상환경과 제시 모듈의 프

로세스를 분리하는 분산처리를 이용하는 방법이 있다.<sup>[6]</sup> 모듈의 확장성과 재사용성을 높이기 위해서는 시스템과 각 모듈간의 네트워크 상에서의 통신규약을 통일하며<sup>[7]</sup>, 동기화에서의 시간지연은 통신대역폭의 확장이나 적절한 통신방법의 선택을 통해서 개선될 수 있다.<sup>[8]</sup>

이와 관련하여 본 논문은 가상환경시스템에서의 운동감 제시 모듈의 디자인에 있어서 고려되어야 할 사항에 대하여 고찰하며, 이전 외부 모듈 연결에 대한 논의를 정리하고 새로운 연구방향을 제시토록 한다.

## 2. 힘 피드백 제어기의 조건

### 2.1. 식별역(識別閾)

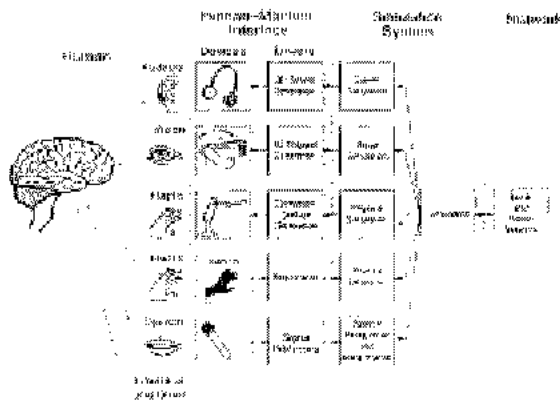


Figure 1. Human Computer Interface

오래 전부터 감각에 대한 연구가 수행되어 왔다. 1958년 Fechner는 Weber의 이론을 바탕으로 감각의 강도는 자극의 대수에 정비례한다는 ‘Weber-Fechner의 정신물리 법칙’을 수학적으로 도출하였다.<sup>[6]</sup> 가상현실 분야에서는 Fechner의 정신물리학(psychophysics)의 응용을 통해 인간 특성치(human factor)를 정의하고, 외부 모듈의 물리적인 최소 필요 사양을 정하였다.<sup>[7]</sup>

정신물리학에서 실험적으로 구해진 인간의 식별역 중 인간이 느끼는 운동감과 관련된 운동발단신경(Mechanoreceptor)의 감각영역은 다음과 같다. 루피니 소체(Ruffini corpuscle)의 경우 0~10Hz 정도의 정적인 압력 자극을 감지할 수 있고, 속도감을 느끼는 마이스너 소체(Meissner corpuscles)의 경우 20~50Hz의 자극 주파수를 가지는 속도를 감지하며, 파니

시안 소체(Pacinian corpuscle)의 경우 100~300Hz의 가속도와 진동의 자극주파수를 감지한다.<sup>[7]</sup>

운동감 제시를 위해서는 촉각에 대한 식별 역뿐만 아니라, 공간에서의 식별 분해능 역시 고려되어야 할 특성이다. 인간은 관절부의 근육과 힘줄에 의해 운동과 위치를 감지할 수 있으며, 관절 부위마다 식별의 분해능이 다르게 나타난다. 팔의 경우 평균적인 식별 분해능을 Table 1.에 나타내었다. 힘의 경우 평균적으로 0.1N 단위로 힘의 차이를 분별해 낼 수 있다.<sup>[7]</sup>

Table 1. Average sensitivity for various arm joints

Joint	Finger	Wrist	Elbow	Shoulder
Sensitivity	2.5°	2.0°	2.0°	0.8°

또 하나 고려되어야 할 특성은 감각에 있어서의 연속적인 인식한계이다. 이는 가상현실의 자극의 갱신을 정하는 데 있어서 중요한 특성이다. 예를 들어, 촉감 감지의 경우 5msec 정도의 인식한계를 가지며, 관절에서의 위치 테이터의 연속은 20msec정도의 인식한계를 가진다. 참고로 눈의 경우 25msec이 인식한계이다.<sup>[7]</sup> 인식한계와 갱신율과의 관계는 2.2.2.절에서 구체적으로 논하고자 한다.

## 2.2 Human Interaction Devices (HID)

### 2.2.1. HID의 분류

인간과 가상현실과의 상호작용을 위해서는 다양한 감각에 대한 체계적인 장비가 필요하다. 이러한 장비를 이 논문에서는 HID (Human Interaction Device)라고 명명하며, 가상현실에서의 좀더 체계적인 운동감 재현을 위해 다음과 같이 분류한다.

우선 입·출력에 의해서만 구분을 하면, 사용자의 입력만을 받는 입력기기로는 휠체어, 조이스틱이 있으며, 가상환경으로부터의 출력만을 나타내는 기기로는 windmaker, 모션플랫폼(motion platform) 등이, 입력과 출력을 동시에 반영하는 장비로는 힘 피드백 조이스틱(Force-feedback Joystick), 암마스터(arm master) 등이 있다.<sup>[9]</sup> 이러한 동적인 HID는

주기적인 갱신율을 가지며 하며, 각 기기마다 갱신율의 차이에 의해 다음과 같은 분류가 가능하다. 빠른 갱신율을 가지는 촉각 위주의 Haptic feedback controller와 비교적 낮은 갱신율을 가지는 관절운동 위주의 Kinesthetic/proprioceptive feedback controller가 있다.<sup>[7]</sup>

햅피드백의 제시를 위한 가상환경의 운영에 있어서는 재구성성(reconfiguration)과 효율성을 높이기 위해서는 위에서 논의한 HID의 성격을 잘 파악하여 시스템을 구성하는 것이 중요하다.

## 2.2.2 HID의 조건

HID가 가상현실 서버와 인간 사이에서 제공해야 하는 기본적인 고려 사항은 전체적인 응답 성능(responsibility), 최대 힘의 상도(magnitude)와 자극 갱신율(bandwidth), 힘의 분해능(resolution), 위치 추적의 정확성(accuracy) 등이다.<sup>[7]</sup> 그렇지만 단순하게 각 기기들이 빠른 갱신율과 높은 힘의 상도, 세밀한 분해능을 가진다면, 가상현실 시스템에서의 프로세스 할당이 높아지고, 각 모듈과의 동기화에 있어서도 많은 문제를 야기하게 된다.

앞서 제시한 인간 특성과와 HID의 분류를 이용하여 디바이스 서버와의 통신에 있어서의 갱신율을 효율적으로 관리한다면 이러한 문제를 해결할 수 있으리라 생각된다. 요컨대 인간의 식별역을 벗어나는 역치에 대해서 HID가 높은 성능을 만족할 필요는 없다. 응답 성능은 실시간을 보장하기 위해 인식한계 이전에서 지연시간을 만족시키면 되고, 분해능 역시 식별역보다 높은 분해능을 가져야할 이유가 없다. 한 예로 Kinesthetic/proprioceptive feedback controller의 경우, Rosenberg<sup>[6]</sup>는 50Hz 이상의 갱신율을 요구하였다. 그러나 Howe와 Kontarinis의 실험<sup>[6]</sup>에서는 8Hz만으로도 충분하다고 결론짓고 있다. 한편, Richard<sup>[10]</sup>의 실험에서는 15Hz의 force feedback loop를 이용해도 좋은 결과를 얻었다고 보고하였다.

이와 같이 햅 피드백 상비는 인간의 감각영역을 고려하여야 한다. 손에서의 촉각을 생밀하게 느끼기 위해서는 감각기의 높은 갱신율을 만족시켜야 한다. 하지만 힘을 느끼는 근

육의 반응기는 낮은 갱신율을 가지더라도 인간의 감각영역을 만족시킨다.

Ellis<sup>[11]</sup>는 촉각과 햅 피드백의 인터페이스 충족 조건을 Table 2에서 보여주고 있다. 본 연구에서 사용한 햅제시 조이스틱의 경우 갱신율을 30Hz로 하고 있지만, 시각과의 이질감을 느낄 수 없었다.

Table 2 Tactile/force feedback interface requirements

	Time Delay	Band width	Resolution	dynamic range	signal/noise
Control	10msec	100Hz	discrete: 1~4bits/dof continuous: 10bits/dof	20N at DC to 1N at 10Hz	200:1
Tactile Sensing	5msec	1~10Hz	10 ~100micro vibration 1~2mm spatial	8bit	200:1 RMS
Force Sensing	20msec	50~100Hz	0.1N	20N at DC to 1N at 10Hz 6bit 1~10cm	64:1 RMS

## 3. EMI Model

### 3.1. EMI의 개요<sup>[9]</sup>

EMI(External Module Interface)는 가상현실 시스템 서버와 외부모듈 사이의 소프트웨어 인터페이스이다. 이와 같은 집단은 서버에서의 최소한의 키네틱 프로세스와 디바이스 서버 내에서의 단일 인터페이스 메커니즘을 제공함으로써 다중 감각의 가상현실 시스템의 최적화된 구성을 가능하게 하는 데 있다.

EMI는 다양한 기능을 제공하는 모듈들이 일관된 방법으로 키네틱에 안정적으로 접근하기 위한 인터페이스를 제공한다. EMI는 시스템이 동작 중에도 외부모듈의 연결, 제거 또는 교체가 가능하도록 디자인되며, 임의의 모듈에서 발생한 오류가 전체 시스템에 영향을 주지 않도록 보호 메커니즘을 지원한다. 또한, EMI는 네트워크간 통신 메커니즘을 기반으로 하기 때문에 외부 모듈이 지역 또는 원격 컴퓨터 상에 위치할 수 있다. 따라서, EMI는 다양한 외부 모듈을 가지는 분산 처리 환경의 구성은

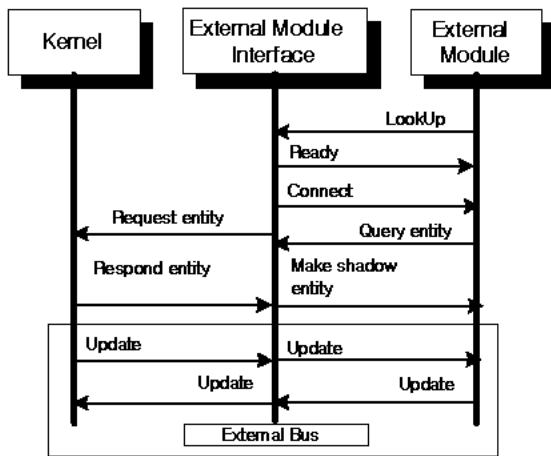


Figure 2. External Module Interface

지원하여 시스템 부하를 분담하여 전체 시스템의 성능을 향상시킬 수 있다. 또한, 개별적인 개발 환경과 시스템 환경에서 독립적으로 개발이 가능하다. 이와 같은 개념은 이후 Device Server와 연관되는 내용이기도 하다.

앞서 얘기한 HID 특성에 따라 EMI의 접근 모드를 구분하면 읽기 전용, 쓰기 전용, 읽기와 쓰기 등이 있다. EMI의 개념을 도입하여 출력 인터페이스의 확장을 위하여 3차원 사운드 생성기, 입력 인터페이스의 확장을 위하여 음성 인식, 손짓 인식, 태이터 알러브, 휠체어, 자전거 등의 HID를 개발한 바 있다. HID를 개발할 때에는 역시 각 모듈 특성에 맞는 분해능과 역치를 고려해야 한다.

EMI의 응용은 프로세스 요구가 높은 컴퓨터 비전이나 원격, 현실감의 구현에 적합하다. 분산 처리 환경에서 프로세스 처리의 분담이나 파이프라인을 형성하여 성능을 향상시킬 수 있다. EMI에 의해 생성된 객체는 각기 하나의 외부모듈만을 관리하며, 네트워크 프로토콜을 통하여 도착한 정보를 일차 해석하여 디바이스를 제어하기 위한 데이터와 객체와 관련된 데이터로 구별한다. 제어를 위한 정보는 외부 모듈의 속성 갱신을 감지하여 다시 가상 환경 시스템 서버에 전달하며, EMI는 변경된 속성 값에 대한 콜백 함수를 호출한다.

외부 모듈은 시스템이 추가적으로 필요한 기능을 제공하며 가상현실 시스템 서버와는 독립적인 프로세스이다. 즉, 응용 분야에 따라

요구되는 다양한 기능을 외부 모듈로 정의하여 시스템에 포함한다.

### 3.2 Using DirectInput in DirectX.

이러한 EMI의 체계적인 운영을 위하여 DirectX라는 멀티미디어 라이브러리를 사용하게 된다. Windows 운영체제에서는 msec 단위의 신뢰할 수 있는 제어가 어려우므로 하드웨어를 직접 조작할 수 있는 라이브러리의 사용이 필요하다. 이런 이유로 Microsoft사의 멀티미디어 라이브러리인 DirectX를 이용하게 되었다.

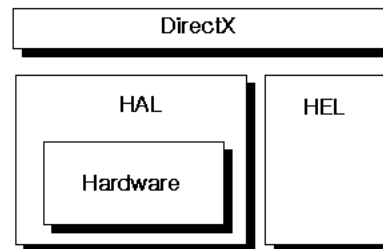


Figure 3. DirectX Architecture

가 외부모듈의 관리에 있어, 동적인 외부모듈을 관리하기 위하여 DirectInput을 이용하여 HID를 관리하게 된다. DirectInput은 하드웨어를 직접적으로 제어하고, 하드웨어의 자원을 독점하여 운영체제와의 독립성을 유지한다.

DirectX는 하드웨어적인 function을 수행하는 HAL(Hardware Abstraction Layer) 계층과 소프트웨어적인 하드웨어 function을 지원하는 HEL(Hardware Emulation Layer) 계층으로 나뉘어서 하드웨어를 관리하게 되는데, 이를 통해 가상현실에서의 현실감을 충족시킬 수 있는 HID제어를 구현한다.<sup>[12]</sup> 웹 제어 조이스틱의 경우 DirectX를 이용하여 EMI를 구성하였다.

### 3.2 Windows Driver Model

DirectX를 이용하는 데 있어서의 문제점은 하드웨어의 메모리번지를 DirectX의 규약에 맞도록 디자인해야 하는 데 있다. 이는 이후에 외부모듈들이 다른 Application(게임 등의 상용프로그램)에서 융통성 있게 연결될 수 있도

를 지원한다.

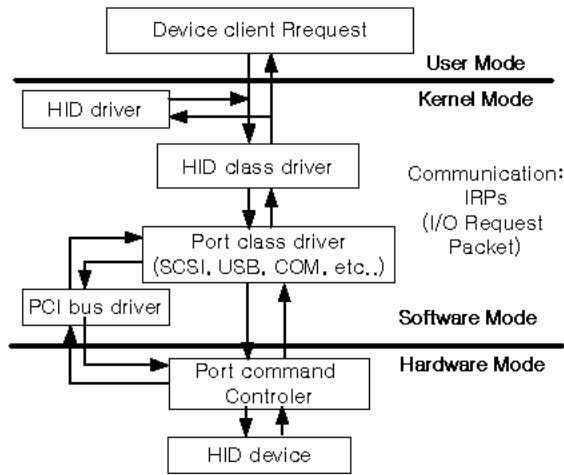


Figure 4. WDM for EMI

## 4. Device Server Model

### 4.1. Client Server Model<sup>[4]</sup>

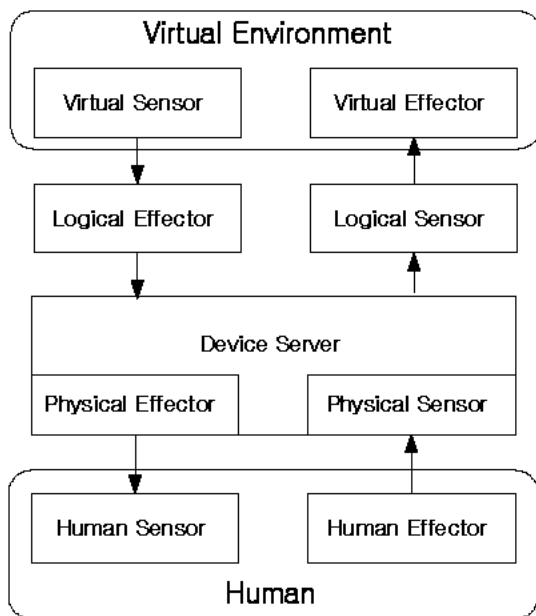


Figure 5. Device Server Model

가상현실 시스템을 개발하는 데 있어서 또 다른 문제점은, 개발 단계에서의 프로그램 부피가 커지게 되며, 이로 인해 개발자 한 명이 담당하여야 되는 프로그램 코드량이 커지게 되는 것이다. 다수의 프로그래머가 개발할 경우에도 각 프로그램의 코어 코드에 대한 보호가 없으므로 개발 도중 야기될 수 있는 위험

부담이 커지게 된다. 또한, 하나의 시스템이 모든 일을 한다면, 하나의 시스템 자원이 담당하여야 하는 프로세스 양이 커져 시스템의 전체적인 성능도 저하하게 된다.

분산처리를 통해 연계 되는 또 다른 이점은 서로 다른 플랫폼끼리의 연결이 가능하다는 것과 시스템의 확장과 계구성의 용이, 그리고 이질적인 시스템 자원을 사용하는 것이 가능하다는 점이다. 따라서 개발 단계에서 서버와 클라이언트 모델을 도입하여, 각 부분이 시스템의 클라이언트로서 동작하여 서버와의 연결이 성립된 경우에는 통합 제시기의 역할을 담당하고, 연결이 성립되지 않은 경우에는 혼자 자신의 역할을 수행하도록 구성한다.

디바이스 서버는 외부 모듈별로 형성될 수 있으며, 혹은 디바이스의 최적값에 맞도록 다수의 디바이스에 대한 디바이스 서버가 될 수 있다. 이러한 디바이스 서버의 디바이스의 개수는 HID의 특성에 맞는 프로세스 량에 의해 결정되며, 서버 시스템과의 통신 대역폭에 맞도록 고려되어야 한다.

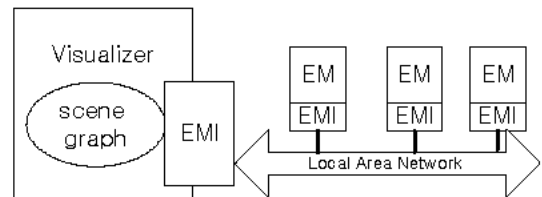


Figure 6. EMI application system

### 4.2. 동기화 통신 구조

다양한 감각 제시를 위해서는 감각에 맞는 통신 방법의 수립이 필요한데, 이미 논의된 바와 같이 감각 제시 정보는 감각에 따라서 제시되는 정보의 양이나 정보의 갱신율이 상이하다.

이를 위해 각각의 감각에 맞게 통신 방법을 수립하는 것이 좋으며, 표준적인 방법으로는 높은 갱신율일 경우 LAN을, 낮은 통신일 경우 시리얼 통신을 근간으로 하고 있다. 하지만 분산된 모듈의 확장성과 계구성의 용이성을 위해 LAN을 이용한 네트워크를 구축하며, 빠른 갱신율을 요구하는 경우에 한해서는 Client-Server가 아닌 Peer to Peer 방식의 통신 방법을 이용한다.

### 4.3. Latency Critical Path

이제 앞서 고찰했듯이 인간의 감각은 어느 정도의 지연이 있더라도 이를 인지하지 못한다. 대부분의 경우 이러한 지연은 역학적인 관성에 의해서 하드웨어의 시간 지연이 대부분을 차지하지만, 시스템 전체에서의 지연을 최대한 억제하기 위해서 최소의 path를 설정할 필요가 있다.<sup>14)</sup>

모든 감각 제시기와 입력기를 서버에 연결하는 방법은 통일성이 있어서 좋으나, 입력기의 정보가 제시기에 빨리 전달되어야 하는 경우, 서버와 클라이언트 모델이 비합리적이다. 입력기의 정보가 서버를 통하여 필요로 하는 제시기에 전달되기 때문에 Latency가 증가되고 따라서 시각 정보의 갱신이 늦어지는 현상이 일어난다. 이러한 path에는 peer to peer 모델이 유리하며, 분산된 path에 대한 중앙 집중적인 모니터링이 필요하다.

## 5. 결 론

가상현실 구축에서 시스템이 확장성과 재구성 가능성을 지원하지 못한다면 응용분야에서 쉽게 도입하기가 힘들며, 개발 과정에도 많은 부담이 가중된다. 또한, HID 기기들이 현실감 재현을 위해 제 역할을 충분하게 하지 못한다면 가상현실에서의 질은 떨어지게 된다.

본 논문은 가상현실 시스템의 외부모듈 사이의 전체적인 인터페이스에 대하여 제안하였다. 여기에는 HID가 만족해야 할 사항에 대한 고찰과 이 디바이스들이 최적의 상황에서 운영되도록 하기 위한 EMI 모델을 제시하고 가상현실 서버와의 위계를 제시하였다.

위의 개념을 2001년 경주 문화엑스포에서 전시한 제어용 힘제시 조이스틱을 활용하여 좋은 결과를 얻었다. 이후 좀더 다양한 외부모듈을 연결하고, 각 외부 모듈의 전체적인 성능 향상에 대한 연구가 필요하다.

## 참고문헌

1. 이계호 외, "운동감 재현을 포함한 실시간 인터랙티브 차량 시뮬레이터 통합에 관한 연구," HCI 2001 학술대회 pp. 731-736, 1999

2. 박창훈, 박경동, 고희동 "가상현실 시스템을 위한 외부모듈 인터페이스의 개발" HCI 학술대회 논문집, pp. 351-355, 1999.
3. 정웅철, 민두기, 송재복, 김용일 "양방향 제어를 이용한 가상환경에서의 팔운동감 제시" 한국감성과학회 추계학술대회, pp 97-100, 1999.
4. 진종욱, 고희동, "시각 제시에 근거를 둔 다중 감각 제시 통합 방법론," 한국감성과학회 추계학술대회 논문집, pp. 185 -188, 2000.
5. 진종욱, 고희동 "감성공학을 지원하는 soft-real time 통신프로토콜의 설계" 한국감성과학회 추계학술대회 논문집, pp. 92-96, 1999.
6. Gescheider, G.A., Psychophysics Method, Theory, and Application, 2nd Edition, Lawrence Erlbaum Associates, pp. 7-11, 1985.
7. Burdea, G.C., Force and Touch Feedback for Virtual Reality, Wiley-Interscience Pub., pp. 13-38, 1996.
8. Rosengerg, L., "How to Assess the Quality of Force-Feedback systems." Technical Report, Immersion Co., Santa Clara, CA, 1995.
9. Richard, P., Burdea, G. et al., "Effect of Frame Rate and Forec Feedback on Virtual Object Manipulation," Presence-Teleoperators and Virtual Environments, Vol. 5, pp. 95-108, 1996.
10. Howe R. and Kontarinis, D., "Task Performance with a Dextrous Teleoperated Hand system," Proc.of SPIE, Vol. 18, pp. 199-207, 1992.
11. Ellis, S., "Human Engineering in Virtual Environments," Proc. of Virtual Reality World '95 Conference, pp. 295-301, 1995.
12. Donnelly, P., "Inside DirectX," Micorsoft Press, 1998.