

□ 특집 □

가상현실 시스템에서의 이벤트 핸들링 및 행위결정

정 문 렬[†]

◆ 목 차 ◆

1. 단일사용자 가상현실 시스템
2. 다중사용자 가상현실 시스템

3 결 론

요 약

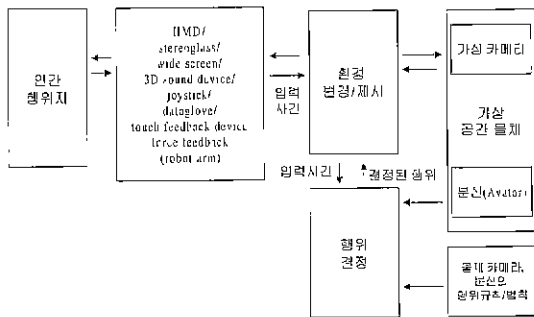
가상현실 (virtual reality) 은 인공현실 (artificial reality), 가상환경 (virtual environment), 가상공간 (cyberspace) 등으로도 불리는데, 컴퓨터에 의해 만들어진 가상의 3 차원 공간으로서 인간이 현실 세계에서 하는 것처럼 이 공간과 상호작용을 할 수 있는 그런 공간을 의미한다. Zeltzer [10] 는 그래픽 기반 시뮬레이션 시스템을 세가지 독립된 축, 즉 자율성 (autonomy), 상호작용 (interaction), 현장임재 (presence) 를 가지고 여러 가지 부류로 분류하였다. 가상현실 시스템은 이 세가지 특성이 모두 완벽하게 존재하는 그런 시스템이다. 완벽한 상호작용은 시각, 청각, 촉각 등을 통해, 가상공간을 인식하는 것과 손, 발 등 신체의 일부분을 이용하여 가상공간에 어떤 행동을 가하는 것을 모두 포함한다. 현장임재는 인간이 가상공간 속에 완전히 들어가 있다는 착각 (몰입감, immersion)을 느낄 수 있어야 가능한 것이다. 자율성은 가상공간내의 물체나 생명체가 외부의 자극과 스스로의 욕구에 의해 자율적으로 움직이는 것을

말하는데 이를 위해서는 물체나 생명체의 행위모델링이 중요하다. 본 글에서는 현실을 감안하여, 3차원 공간과의 상호작용이 있으면 가상현실의 범주에 드는 것으로 간주하고, 실용적인 관점에서 이러한 가상공간을 구축하는데 필요한 요소에 대하여 살펴본다. 가상현실 시스템의 요소기술은 결국 가상현실 시스템 프로그래밍을 보다 효과적으로 그리고 보다 쉽게 도와주는 기술이다. 현장임재를 위해서는 기본적으로 출력 장치 (Headmount Display (HMD), stereoglass, wide screen) 에 관한 기술이 중요하다 [2]. 그러나 본 글에서는 이에 대해서는 다루지 않는다. 입출력 장치를 다루는 구체적인 방법에 대해서도 기술하지 않고 이를 추상적으로 다룬다. 즉 요구되는 기능을 갖춘 입출력 장치가 있다고 가정하고 이들을 이용하여 상호작용을 어떻게 구현할 것인지 그리고 가상공간의 자율성을 위한 행위모델링에 대해 주로 언급하고자 한다. 다시 말하면 인간참여자로부터 발생하는 입력사건 (이벤트)을 처리하는 문제와 입력사건과 관련된 물체의 행위를 결정하는 문제에 대하여 다루고자 한다. 가상현실에 대한 보다 개념적인 소개의 글은 [1, 4,7]을 참고하기 바란다.

[†] 정회원 : 숭실대학교 정보과학대학 컴퓨터학부 조교수

1. 단일사용자 가상현실 시스템

우선 건물내부 워크스루 (walkthrough) 시스템과 같은 단일사용자 가상현실 시스템을 구성하는 요소를 살펴본다. 나중에 기술할 다중사용자 시스템은 단일사용자 시스템을 조합하여 구현할 수 있다. 단일사용자 시스템의 구성은 (그림 1)에 나타낸 바와 같다. 먼저 상호작용, 보다 구체적으로 입력사건 처리 방식에 대해 간단히 설명한다.



(그림 1) 단일사용자 가상현실 시스템

1.1 입력사건 처리 방식

가상현실 시스템은 사용하는 입력장치 (조이스틱, 마우스, 데이터글러브, 헤드마운트 트랙커 등)로부터 일련의 신호 (이를 입력사건 (input event)라 함)를 받아 그 입력사건에 부여된 사용자의 의도를 가상공간에서 실현시키고 그 결과를 출력장치 (헤드마운트 디스플레이, 스테레오 글래스 등)를 통해서 사용자에게 전달하는 프로그램이다. 사용자가 입력장치를 통하여 입력사건을 가상현실 시스템에 보낼 때 어떤 방식으로 하면 좋은가 하는 문제는 인간컴퓨터 상호작용 문제로서 가상현실 시스템의 근본적인 문제중의 하나이다. 그러나 여기서는 일단 입력사건이 발생되었을 때 이를 어떻게 처리하는가 하는 문제에 대하여 살펴본다. 가상현실 시스템은 입력사건을 해석하여

가상공간내에 적절한 행위를 발생시켜야 한다. 이를 사건-행위 연결문제라 하자. 이 문제를 보다 다양하게 그리고 보다 직관적으로 처리하기 위하여 Open Inventor[8,9], WorldToolKit[13], Cyberspace Developer Kit-CDK[11], VR-DECK [12]등의 툴킷이 개발되었다. 그러나 여기서는 보다 간단한 시스템에서 이 문제를 어떻게 처리하는가를 살펴본다. 툴킷에서 새롭게 도입된 방법은 입력사건을 환경을 구성하고 있는 물체 (object)에 보내고 이 물체내에 정의되어 있는 방법에 의해 사건을 처리하는 것이다. 전통적인 방법은 사건처리함수를 프로그래머가 작성하고 입력사건을 이 함수에 보내어 처리하게 한다. 본 글의 관점에서는 두 방법은 크게 다르지 않다. 따라서 보다 전통적인 방법만 기술하고자 한다. 다중 윈도우를 지원하는 시스템 (예, 윈도우즈 프로그래밍을 지원하는 시스템)에서는 사건-행위 연결문제를 해결하기 위해 특별한 제어구조 (control structure)를 지원하는 것이 보통이다. 사용되는 제어구조는 콜백함수 (callback functions) 방식, 사건주도 (event-driven) 방식, 그리고 사건탐색 (event polling)방식으로 나눌 수 있다. 한 시스템에서 한 가지 이상의 제어구조를 지원하는 경우도 있다. 각각의 제어구조에 대해서 간단히 설명하고 이들이 가상현실 시스템을 구현하는데 어느 정도 적합한지 평가하고자 한다.

프로그래머가 작성한 함수를 시스템이 호출할 때 이를 콜백함수라 한다. 콜백함수는 일반 함수와 구분하게 위해 특별한 방식으로 정의된다. 콜백함수의 언어적 의미는 "거꾸로 부르는" 함수라는 뜻이다. 프로그래머가 작성한 함수에서 시스템이 제공하는 함수를 호출하는 것이 자연스러운 방향이라면, 시스템에서 프로그래머가 작성한 함수를 호출하는 것은 거꾸로 된 방향이기 때문이다. 사용자가 콜백함수를 정의할 때, 그 함수에

서 하는 일과 그 함수가 호출될 조건, 즉 특정 입력사건을 지정한다. 작성된 프로그램이 실행될 때, 특정 입력사건이 발생하면 시스템이 이 입력사건을 인식하여 이에 대응되는 콜백함수를 직접 호출한다. 이 방법은 프로그래머가 입력사건을 인식하고 처리하는 과정에 신경을 쓰지 않아도 되는 장점이 있다. 그러나 이 방법은 입력사건과 행위와의 대응관계가 단순한 일대일 관계라는 전제가 있을 경우에만 사용할 수 있다. 만약 두 개 이상의 입력사건에 하나의 행위가 대응된다거나 특정행위의 발생여부가 입력사건뿐만 아니라 가상세계의 상태와 관계가 있다면 이 방법을 쓰기가 곤란하다[3].

사건주도방식은 프로그래머가 하는 일이 콜백방식보다 조금 많다. 프로그래머는 여러 가지 입력사건을 처리하는 사건처리함수를 작성하고, 또한 입력사건의 사건 큐(event queue)에서 입력사건을 하나씩 빼내어 이를 시스템에 보내는(dispatch) 프로그램을 작성한다. 시스템은 사용자 프로그램으로부터 입력사건을 받으면 프로그래머가 작성한 사건처리함수를 호출하여 이 사건에 대응되는 행위를 실행하게 한다. 사건처리함수는 여러 가지 사건을 처리한다는 점에서 위에서 말한 콜백함수와 다르지만 시스템에 의해 호출되는 프로그래머가 작성한 함수라는 점에서 콜백함수이다. 이 방식에서도 입력사건을 인식하여 사건 큐에 넣는 일은 시스템의 몫이고 사용자 프로그램은 사건 큐에서 입력사건을 꺼내오기만 하면 된다. 사건주도방식은 사건 큐가 프로그래머의 제어하에 있다는 점이 콜백함수방식과 다르지만, 전체적으로 볼 때 크게 다르지 않다. 따라서 콜백함수방식의 단점이 사건주도방식에도 그대로 나타난다. 한편, 사건주도방식에서도 시스템이 사건 큐에 입력사건을 넣지 않고 직접 사건처리 콜백함수를 호출할 수 있다.

콜백함수 방식과 사건주도방식은 입력사건의 발생여부를 사용자 프로그램에서 직접 체크하지 않고 시스템에서 체크하여 사용자 프로그램에게 알려준다. 사건주도방식에서는 사용자 프로그램이 사건 큐에서 입력사건을 꺼내오는 무한 루프를 가지고 있지만 사건 큐를 체크하는 함수가 사건 큐가 비어있다는 것을 인식하는 순간 운영체제가 그 사용자 프로그램을 휴식상태로 만든다. 따라서 입력사건의 발생여부를 사용자 프로그램에서 일일이 체크하지 않는다고 말할 수 있다. 따라서 콜백함수방식과 사건주도방식에서는 사용자 프로그램이 입력사건을 기다리면서 CPU 시간을 낭비하는 일이 없다. 그러나 하나 이상의 입력장치로부터 입력사건을 동시에 받아들여 처리하기 어렵다. 그리고 행위의 발생이 입력사건뿐만 아니라 현재의 상태에 종속되어 있는 경우(이를 행위의 상태종속성이라 함) 이를 처리하기 어렵다. 이 두 문제는 사용자와 가상공간과의 상호작용이 복잡해지는 가상현실 시스템에서 자주 발생하는 문제이다. 하나 이상의 입력사건을 동시에 고려하는 문제는 전투게임 등에서 이미 문제가 되고 있다. 예를 들어, 비행기를 타고 특정방향으로 비행하면서 총을 쏘려면 두 개의 입력장치를 이용해 두 개의 입력사건을 동시에 발생시켜야 한다. 콜백함수방식과 사건주도방식은 그 제어구조의 특성상 두 개의 입력사건을 동시에 인식하여 처리하기가 매우 어렵다. 이런 문제를 해결하기 위해 게임에서는 사용자 프로그램에서 직접 사건 큐를 탐색하는 사건폴링방식을 쓰는 경우가 많다. 윈도우즈 프로그래밍 언어에서도 사건폴링을 위한 함수를 제공하고 있다. 사건폴링방식은 사용자 프로그램이 입력사건을 기다리면서 CPU 시간을 낭비하는 단점이 있다. 그러나 이 방식은 사건-행위간의 관계를 특정제어구조하에서 도식적으로 기술하는 것이 아니라 사용자 프로그램에서 자유롭게

게 기술할 수 있다. 따라서 다중입력사건과 행위의 상태종속성을 처리할 수 있는 여지가 있다. 콜백함수방식과 사건주도방식이 폴링방식의 단점을 극복하기 위해 제공된 입력처리방식이지만 위에서 말한 것처럼 복잡한 상호작용을 처리하는데는 단점이 있다.

가상현실 시스템에서는 외부의 입력사건을 행위를 결정짓는 여러 요소중의 하나로 생각해야 한다. 행위결정은 매우 복잡한 실시간 추론문제로 귀착이 된다. 따라서 입력사건과 행위를 일대일로 연결하는 콜백함수방식과 사건주도방식은 사용하기가 어렵다. 그렇다고 사건 폴링방식을 사용해야 된다는 것은 아니다. 입력을 탐색하는 독립적인 프로세스를 두고 이것이 입력사건을 행위결정프로세스에게 보고하는 방식을 취하면 행위결정프로세스가 입력사건을 탐색하면서 시간을 낭비하는 문제를 해결할 수 있다.

1.2 행위결정을 위한 행위모델링

가상공간을 구축하려면 우선 공간에 놓일 물체와 분신(가상공간에서 사용자를 대표하는 물체)을 모델링하는 일이 필요하다. 물체의 모델링은 형상모델링과 행위모델링으로 나눌 수 있다. 행위모델링은 물체가 외부로부터 어떤 자극(힘, 위협, 명령 등)을 받았을 때, 어떻게 반응할 것인가를 기술하는 것이다. 즉 행위결정을 위해서 행위모델링이 필요하다. 일반적으로 행위는 두 레벨로 나누어 모델링되는데, 상위레벨의 행위는 스크립트 언어를 이용하여 모델링하고, 하위레벨의 행위는 운동학적/동역학적으로 기술한다. 스크립트레벨의 행위기술은 물체가 주어진 상황에서 어떤 행위를 할 것인가를 정하는 것이다. 만약 우리가 가상공간의 물리적 상태와 움직이는 물체나 사용자의 분신의 내면상태를 아주 정확하게 알고 있다면 주어진 상황에서 어떤 행위가 일어날지 매

우 정확하게 예측할 수 있을 것이다. 그러나 이것은 불가능하다. 따라서 주어진 상황에 대응되는 행위를 결정하는 문제는 어느 정도 작위적이다. 예를 들어 사람이 자전거를 타고 가다가 돌부리에 부딪쳐서 넘어질 때, 어느 방향으로 어떻게 넘어질 것인가를 정확히 예측하려면 너무 많은 것을 알아야 한다. 따라서 물론 필요에 따라 보다 정확하게 할 수도 있지만 상식적으로 납득되는 한도내에서 특정방향으로 특정방식으로 넘어진다고 정할 수 있다. 하위레벨의 행위모델링은 이렇게 발생될 행위가 구체적으로 결정이 되었을 때 이 행위를 생성하는 방법을 기술한다. 이를 위해서는 상위레벨 행위모델링의 결과가 원하는 행위를 물리적인 계산방법으로 결정할 수 있을 만큼 구체적이고 정량적(quantitative)이어야 한다. 즉 상위레벨의 행위모델링의 최종결과는 운동학적 또는 동역학적 행위기술이어야 한다. 운동학적 행위기술은 해당물체의 속도나 가속도를 기술하고 동역학 법칙 레벨의 행위기술은 질량, 탄성계수 등 해당 물체의 물리적 성질, 이 물체가 받는 힘, 만족해야 할 운동법칙, 그리고 다른 물체와의 관계에서 만족해야 할 제약조건(constraints)을 기술한다. 운동학적 또는 동역학적 행위기술을 만족하는 행위를 실시간에 생성하는 문제는 매우 어려운 문제로서 연구가 진행되고 있는 분야이다. 한 방법은 모션 캡춰등을 이용하여 미리 동작을 만들어 놓고 이를 주어진 행위기술을 만족하도록 변형하는 방법이다[5]. 그런식으로 하여 물체들이 현재 프레임에서 다음 프레임까지 어떻게 움직여야 되는지 결정이 되면 이를 환경변경/제시 모듈에 보내어 환경(장면 그래프)을 실제로 변경시키고 그 결과를 인간참여자에게 제시하게 한다. 물체의 행위와 관련된 성질과 법칙을 물체의 모양처럼 물체의 한 특성(attribute)으로 보고 물체를 모델링하는 것이 좋다. 이는 객체지향의 원리

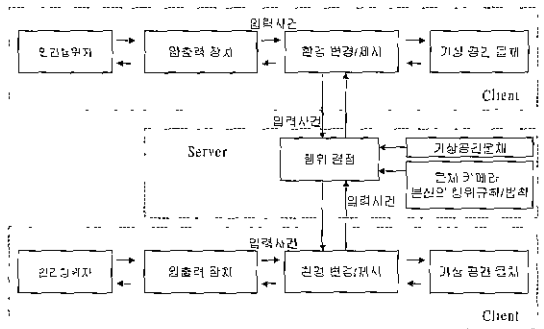
를 일관되게 적용하는 것으로서 공간이 복잡해질 때 공간의 물체의 성질을 수정, 보완할 때 체계적인 모델관리를 가능하게 해주기 때문이다. 모델링 물체 모델링은 가상현실 시스템뿐 아니라 3차원 게임과 같은 일반 3차원 그래픽스 응용시스템에서도 필요한 일이다.

가상공간내에서 행위를 발생시키는 것과 더불어 이를 인간참여자에게 얼마나 잘 제시하느냐가 중요하다. 가상카메라를 입력사건이나 발생하는 행위에 적합하게 제어하는 문제도 중요하다. 따라서 그림 1에서 행위결정 모듈이 카메라 제어 규칙[6]을 사용하도록 하였다.

2. 다중사용자 가상현실 시스템

다중사용자 가상현실 시스템은 다수의 사용자가 동일 공간을 공유하는 시스템으로서 3차원 채팅이나 네트워크 게임등에 활용될 수 있다. 다중사용자 가상현실 시스템을 구현하기 위해서는 네트워크, 멀티 태스킹, 장면 그래프 일관성 유지 문제 등 여러 가지 기술적인 문제들이 해결되어야 한다. 그러나 이 글에서는 입력사건처리와 행위결정문제에 국한시켜 이 문제를 살펴보고자 한다. 그림 2에서 보듯이 각 클라이언트는 동일한 가상환경을 각각 가지고 있다. 장면을 렌더링하여 사용자에게 제시하는 일은 클라이언트에서 국지적으로 해야 되기 때문이다. 그리고 각 사용자가 공간상의 물체에 영향을 주는지를 체크해야 하기 위해서는 장면 그래프를 국지적으로 가지고 있어야 하기 때문이다. 동일한 공간을 공유하고 있는 다중사용자로부터 입력사건이 들어 올 경우, 이에 대응되는 행위는 다수의 입력사건을 동시에 고려하여 결정되어야 한다. 이를 구현하는 한 방법은 각 클라이언트가 다른 모든 클라이언트로부터 입력사건을 모두 전달받아 이에 대응되는 행위를

결정하는 것이다. 즉 각 클라이언트에서 행위규칙을 이용해 행위를 결정하고 결정된 행위를 이용해 환경을 변경한다. 또 한 방법은 (그림 2)에 보인 바와 같이 각 클라이언트에서 입력사건을 서버에 전달해 주고 서버에서 행위규칙을 이용해 행위를 결정하고 결정된 행위를 각 클라이언트에 전달해 환경을 변경하는 것이다.



(그림 2) 다중 사용자 공유 가상현실 시스템

두 방법 다 장단점이 있다. 보통의 경우 서버의 계산 능력이 클라이언트에 비해 월등히 좋기 때문에 계산이 많이 요구되는 행위 결정문제를 서버에 넘기는 방법이 합리적인 것이라 생각된다. 클라이언트가 서버로부터 받는 결정된 행위는 결국 해당 물체의 새로운 위치와 방향 데이터이기 때문에 서버와 클라이언트간에 주고받는 데이터의 양은 그리 크지 않다. 한편, 각 사용자는 가상공간의 특정부분만을 보고 있기 때문에 시야밖에 있는 물체의 행위를 무시해도 된다. 따라서 각 클라이언트에서 행위를 결정할 때는 시야 안에 있는 물체만 고려할 수 있는 장점이 있는데, 서버에서 하게 되면 가상공간내의 모든 물체의 행위를 고려하거나 아니면 각 사용자의 시야를 모두 합한 시야 내에 들어가는 물체를 고려해야 한다. 그리고 가상공간의 물체를 공유되는 전역 물체와 공유되지 않는 국지 물체로 나누는 경우에는 각

클라이언트에서 행위를 결정하는 것이 편리하다.

3. 결 론

가상현실 시스템을 구현할 때 필요한 입력사건 처리 방식과 행위결정방식에 대하여 간단히 살펴 보았다. 전통적인 방식으로 입력사건과 행위를 일 대일로 대응시키는 것은 상호작용이 단순하지 않은 가상현실 시스템에 적합하지 않다는 결론을 얻었다. 그리고 입력사건처리는 행위결정이라는 복잡하고 어려운 문제와 긴밀히 연결되어 있음을 알았다. 이 두 문제를 해결하기 위해서는 전통적인 컴퓨터 그래픽스 프로그래밍 스타일을 과감히 벗어나려는 시도가 필요하다고 판단된다.

참고문헌

[1] 원광연, 전산학으로서의 가상현실, 정보과학회 지 제 15권 제 11호, p. 5-13.

[2] G. Burdea and P. Coiffet, Virtual Reality Technology, John Wiley & Sons, Inc, 1993.

[3] A van Dam, VR as a Forcing Function: Software Implications of a New Paradigm, Proc. IEEE Symposium on Research Frontiers in Virtual Reality, pp. 5-8, Oct. 1993.

[4] Ellis, what are virtual environments?, IEEE Computer Graphics and Applications, Vol. 4, No. 1, pp. 17-22, Jan. 1994.

[5] M. Gleicher, Motion Editing with Spacetime Constraints, Proceedings of the Symposium on Interactive 3D Graphics, 1997.

[6] Li-wei He, Michael Cohen, David Salesin, "The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing" SIGGRAPH '96

[7] J. Latta and D. Orberg, A Conceptual Virtual Reality Model, IEEE Computer Graphics and Applications, Vol. 4, No. 1, pp.23-29. Jan. 1994.

[8] P. Strauss, An Object-Oriented 3D Graphics Toolkit, Computer Graphics, 26,2, July 1992.

[9] Josie Wernecke, The Inventor Mentor, Addison-Wesley Publishing Co., 1994.

[10] D. Zeltzer, Autonomy, interaction, and presence. Presence: Teleoperators and Virtual environments. 1(1): 127-132, Winter 1992.

[11] Autodesk Inc., Cyberspace Developer Kit Concepts & Components, CyTechnical Note # 1, CA, 1993.

[12] Dimension Ltd., Amaze User Manual, Bristol, UK, 1993.

[13] Sense8 Co., WorldToolKit Version 1.01, Technical Brief, CA, November 1992.



정 문 렬

1980년 서울대학교 자연대학 계산
통계학과 (학사)
1982년 한국과학기술원 전산학과
(석사)
1982년-1986년 국방관리연구소
(KIDA) 연구원

1992년 University of Pennsylvania, 전산학 (박사)
1992년-1994년 일본주쿠공업대학교 정보공학부 조교수
1994년-현재 숭실대학교 정보과학대학 컴퓨터학부
조교수

관심분야 : 인체 애니메이션, 모션 캡처 기술, 물리기반
시뮬레이션, 가상공간 상호작용, 가상공간
행위모델링