다중사용자를 지원하는 3차원 가상환경 서버의 개발

고명철° 김종혁 정혜원 최윤철
연세대학교 컴퓨터과학과
{zoo, iwannado, leojung, ycchoy}@rainbow.yonsei.ac.kr

Development of 3D Virtual Environment Server supporting Multi-user Participation

Myeong-Cheol Ko° Jong-Hyuk Kim Hye-Won Jung Yoon-Chul Choy
Dept. of Computer Science, Yonsei University

요 약

기존의 웹 기반 3차원 가상환경 분야에 대한 연구는 시각적인 면에서 이전의 웹 컨텐츠에 비해 많은 향상이 있었다. 그러나 내부 객체들간의 다양한 상호작용이나 참여자를 대신하는 아바타(Avatar)의 행위(Behavior)표현에 대해서는 많은 한계를 가지고 있다. 본 논문에서는 아바타를 포함한 가상 객체간의 상호작용과 행위의 유형을 정의하며 이를 지원할 수 있는 가상환경서버를 설계하고 구현한다. 또한, 다중사용자의 참여로 인해 발생할 수 있는 서버의 성능저하를 줄이기 위한 방법론으로서 지역관리 및 메시지 필터링 기법을 제안한다. 구현된 시스템은 가상 쇼핑몰 등의 응용분야에 실제 적용이 가능하다.

1. 서론

최근 웹에 기반 한 3차원 가상환경 분야에 대한 연구가 진행되고 있다. 가상환경을 구축하고 활용하는 데에는 네트워크나 컴퓨터 그래픽과 같은 컴퓨터 기본기술과 가상세계에 존재하는 다양한 객체간의 상호작용을 처리하는데 필요한 가상현실 기술이 접목되어야 할 필요가 있다. 그러나 기존의 연구는 VRML(Virtual Reality Modeling Language) 등을 이용하여 단순히 3차원적인 가상환경을 구축하고 그 공간을 탐색(Navigation)하는 수준에 머물고 있다[1]. 또한, 일부의 연구에서 지원하고 있는 객체간의 상호작용이나 참여자를 대신하는 아바타(Avatar)의 의사 및 행위 표현은 많은 한계를 가지고 있다[2]. 본 논문에서는 가상환경 내의 객체들을 그 특징에 따라 분류하고 아바타를 포함한 가상 객체간의 상호작용의 유형을 정의하여 이를 지원할 수 있는 가상환경서버를 설계하고 구현한다. 또한 3차원 데이터를 기반으로 하는 가상환경 시스템에서는 참여자간 데이터 전송이 서버의 전체적인 성능에 막대한 영향을 미칠 수 있으므로 이의 해결을 위한 방법론을 제안한다. 본 논문에서 제안된 시스템은 실제 가상 쇼핑몰 등의 응용분야에 적용 가능함을 확인하였다.

2. 가상 객체(Virtual Object)

본 절에서는 가상공간 내에 존재하는 다양한 가상객체를 그 특징에 따라 분류해 보고 이들 각각의 행위유형을 분석해 봄으로써 가상환경 서버가 제공해야 할 기능에 대해서 고찰해 본다.

2.1 가상객체의 분류

표 1 가상객체의 분류

가상객체	특징
Avatar	참여자를 대신하는 가상공간 내의 객체로서 일반적으로 지능적인(Intelligent) 특징을 갖는다.
Autonomous Object	자신의 기능 혹은 행위가 컴퓨터 프로그램에 의해 미리 정의되어 있는 객체로서 제한된 범위 내에서의 지능을 갖는다. 참여자와의 상호작용을 위한 자체 데이터베이스를 가지고 있다.
Dynamic Object	일정한 동작을 반복하는 것은 다음의 Animated Object와 유사하나 Avatar에 의해 사용(trigger)된다는 점이 다르다(Interactivity).
Animated Object	단순한 행위만을 반복하는 객체.
Static Object	자신의 행위를 가지고 있지 않은 가상 공간내의 고정된(static) 객체.

본 논문에서는 가상공간 내에 존재하는 객체들을 그 기능 및 특징에 따라 표 1과같이 모두 다섯 가지 유형으로 분류한다.

본 연구는 산업기반기술개발사업(과제번호: 2000-2-0074)의 지원으로 수행되었음.

2.2 가상객체 간의 상호작용

위 2.1절에서 분류한 가상 객체간 상호작용의 유형을 정의해 보면 다음과 같다.

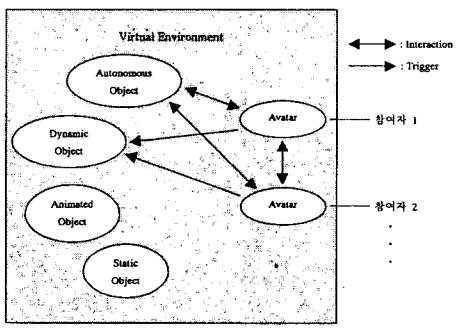

그림 1 가상 객체간의 상호작용

- Avatar vs. Avatar
  Avatar는 참여자의 분신으로서 참여자의 모든 행위나 상호작용은 Avatar를 제어함으로써 수행된다. 따라서 지능적인 수준에서 일반적인 모든 상호작용이 가능하다.
- Avatar vs. Autonomous Object
  Autonomous Object 자신이 가지고 있는 지식의 범위 내에서 Avatar와의 지능적인 상호작용이 가능하다.
- Avatar vs. Dynamic Object
  Avatar의 일방적인 요구에 대해 미리 정해진 Dynamic Object의 반응이 동기화 된다.
- Avatar vs. Animated Object/Static Object
  Avatar와의 상호작용과는 무관하게 가상공간 내에 존재하는 정적인 모든 객체.

2.3 가상객체의 행위(Behavior) 유형

본 논문에서는 기본적으로 지능적인 차원에서의 행위만을 정의한다. 가상객체 중 지능적인 특징을 갖는 객체는 다음의 Avatar와 Autonomous Object이다.

■ Avatar의 행위 유형

Avatar는 지능적인 특징으로 인해 그 행위의 유형이 매우 복잡하고 다양하다. 따라서 모든 응용분야에 일반적인 Avatar의 행위를 유일하게 정의하는 것은 불가능하다. 본 논문에서는 가장 기본적인 Avatar의 행위를 표 2와 같이 분류한다.

표 2 Avatar의 행위 유형

대화(Conversation)	타 Avatar 또는 Autonomous Object와의 대화
항해(Navigation)	가상공간 내의 이동
공간이동(Teleport)	가상공간 내의 특정 위치로 점프(Portal)
선택(Selection)	반응할 수 있는 가상공간 내의 특정 객체를 선택하는 행위
애니메이션(Animation)	Avatar 자신의 신체의 일부를 움직이는 행위

■ Autonomous Object의 행위 유형

Autonomous Object는 해당 응용시스템의 성격을 가장 잘 반영하는 객체라고 볼 수 있다. 이들은 미리 프로그래밍 된 행위와 자신의 데이터베이스를 가지고 Avatar와 상호작용 하게 된다. 응용시스템에 따라서는 사람이 Autonomous Object를 직접 조정할 수도 있는데 이러한 경우 위 Avatar의 행위 유형과 매우 흡사해 진다. 본 논문에서는 이러한 경우는 배제한다.

표 3 Autonomous Object의 행위 유형

대화(Conversation)	Avatar와의 대화
항해 (Navigation)	가상공간 내의 이동
애니메이션(Animation)	Autonomous Object 자신의 신체의 일부를 움직이는 행위

3. 클라이언트 및 서버의 구성

앞서 2절에서 분류한 가상객체의 다양한 행위와 상호작용을 지원하기 위해 본 논문에서는 아래와 같이 모두 다섯 가지의 서버모듈을 구현한다. 그림 2는 본 논문에서 구현한 클라이언트 및 서버모듈의 구성과 클라이언트를 이용한 서버로의 접속과정을 보인 것이다.

■ Login Manager

참여자의 로그 인을 처리하며 입력이 적절한 경우 Region Manager에게 입장을 요청하고 이것이 받아들여 질 경우 참여자는 Client Manager를 할당 받아 가상공간으로 입장하게 된다.

■ Client Manger

Avatar와 가상 객체들간의 다양한 상호작용을 처리한다. 필요할 경우 서버의 Region Manager와 수시로 통신하면서 각 Avatar들에 대한 상태정보(위치, 방향 등)를 갱신하며 가상공간 내부의 전체적인 일관성(Consistency)을 유지한다.

■ Region Manager

모든 가상객체의 위치 및 방향정보와 상호작용의 결과로서 발생하는 다양한 이벤트들을 감지하고 이를 가상공간 내의 모든 객체들에게 전파하여 변경된 내용이 적절히 반영되도록 한다.

■ Event Manager

클라이언트로부터 발생하는 이벤트를 Region Manager의 각 모듈로 적절하게 분기 시키고, Region Manager 내의 각 모듈로부터 발생하는 이벤트를 클라이언트로 중계하는 역할을 한다.

■ Avatar Manager

Avatar의 행위나 대화(Chatting), 애니메이션 등과 관련된 다양한 이벤트들을 처리한다.

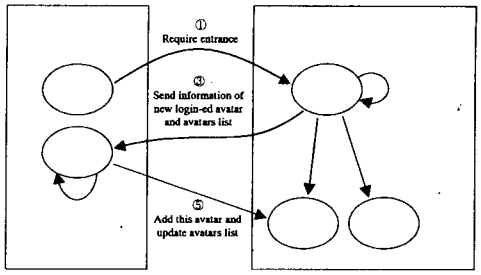

그림 2 클라이언트/서버의 구성 및 참여자 로그인 과정

4. 참여자의 관리

4.1 Login Queue

Login Queue는 참여자의 동시접속을 처리하기 위한 내부 메커니즘이다. 참여자들은 접속한 순서에 따라 해당 프로세스가 Login Queue에 저장되며 Queue의 앞쪽(front)에서부터 하나씩 로그인 작업을 수행하게 된다(그림3). 이 때 첫번째 프로세스의 작업이 모두 완료된 이후에 두 번째 프로세스가 작업을 시작할 수 있다. 각 프로세스의 작업을 병행적(Parallel)으로 처리하지 않는 이유는 가상환경 시스템의 경우 데이터의 규모가 매우 방대하기때문에 초기 데이터 전송 시간이 일반 서버에 비해 더 오랠 수 있으며 만일 이를 병행적으로 처리할 경우 가상환경의 전체적인 일관성을 보장할 수 없기 때문이다.

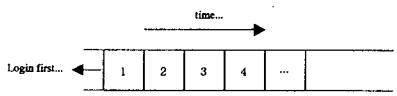

그림 3 Login Queue

4.2 Avatar의 위치 및 방향정보의 추적(Tracking)

각 참여자들은 가상공간 내에서 수시로 위치를 이동하거나 시점(Viewpoint)을 변경시킬 수 있다. 이때 이러한 참여자의 이벤트를 계속적으로 추적하고 그에 따른 새로운 영상을 가상공간내의 모든 참여자들에게 적절히 제시하여야 한다.

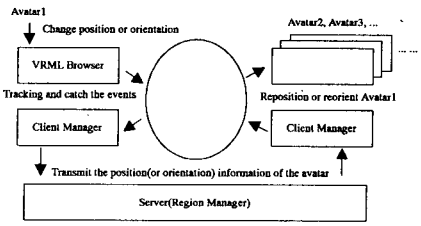

그림 4 Avatar의 위치 및 방향정보의 추적

그림 4는 이러한 추적과정이 Client Manager와 Region Manager 사이에 어떠한 경로로 이루어지는지를 보인 것이다.

4.3 Avatar 이벤트의 처리

각 참여자들은 가상객체와의 상호작용의 결과로서 다양한 이벤트를 생성시킬 수 있다. 이는 자신은 물론 타 참여자의 브라우저에도 동일한

결과가 반영되어야 한다. 그림 5는 참여자의 이벤트를 처리하는 과정을 보인 것이다.

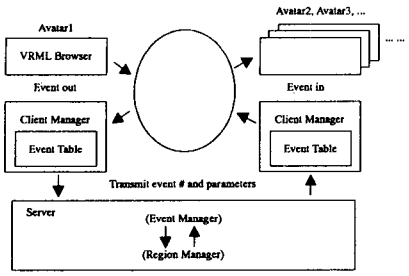

그림 5 Avatar 이벤트의 처리

5. 가상환경의 관리

5.1 지역 관리(Region Management)

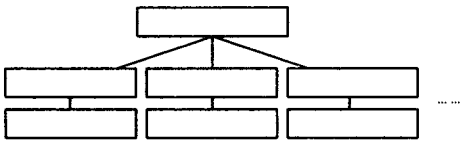

그림 6 계층적 지역관리

대규모의 가상환경을 하나의 지역(Region)으로 관리하는 것은 시스템의 전체적인 성능면에서 바람직하지 못하다. 본 논문에서는 Region Manager를 통하여 전체 가상환경을 여러 개의 하위 지역들로 분할 관리하는 방법을 이용한다[5]. 즉, 그림 6에서와 같이 각 지역별로 Region Manager를 두어 해당 지역을 관리하게 하고 이들간의 통신을 위해서 상위에 Super Region Manager를 두는 형식을 취한다.

5.2 메시지 필터링(Message Filtering)

Extent of Avatar's DOI(Degree Of Interest)

Area1 > Area2 > Area3 > Area4

So,

Time interval of message transmission

Area4 > Area3 > Area2 > Area1

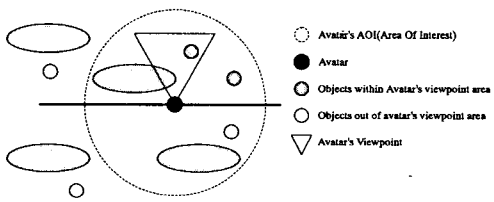

그림 7 AOI(Area Of Interest)를 이용한 메시지 필터링

Message Filtering[4]은 앞서 지역관리 기능과 비슷한 이유에서 본 논문에서 도입한 개념이다. 가상공간 내에서 발생하는 모든 이벤트를 클라이언트 측으로 전송하는 것은 서버에 과중한 부담을 주어 전체적인 성능의 저하를 초래할 수 있다. 따라서 이를 선별하여 불필요한 메시지는 전송하지 않는 것이 메시지 필터링의 기본 원리이다.

그림 7에서 Avatar의 다음 위치는 Area1 지역에 있을 가능성이 가장 높다. 따라서 Area2, Area3, Area4 지역에 있는 Avatar들에게는 메시지를 드물게 전송하고 Area1 지역에 있는 Avatar들에게는 계속적으로 메시지를 전송하게 된다. 또한 Area4 지역에 위치한 Avatar가 갑자기 Area1 지역으로 들어왔을 때 위치나 방향정보에 오차가 생길 수 있으므로 본 논문에서는 이의 보정을 위한 *Keep-Alive Message*를 매 5초마다 일률적으로 모든 Avatar들에게 전송하는 방법을 사용한다.

메시지는 주로 Avatar의 위치 및 방향정보 등으로서 이에 대한 샘플링 율도 서버의 성능에 중요한 인자로서 작용할 수 있다. 본 논문에서는 위치에 대해서는 매 0.3m마다 방향에 대해서는 매 5°마다 각각 샘플링 하는 방법을 사용한다.

6. 결론

본 논문에서는 가상공간 내에서 참여자가 단순히 3차원 공간을 탐색하는 수준이 아닌 가상공간 내 객체들과의 다양한 상호작용이나 의사교환을 지원할 수 있는 3차원 가상환경 서버를 구현하였다. 또한 많은 수의 참여자로 인해 발생할 수 있는 서버의 성능저하 문제를 해결하기 위해 가상환경을 효율적으로 관리할 수 있는 방법론을 제안하였다. 본 논문은 실제 네트워크 상에서 다자(multi-user) 참여를 기본으로 하는 가상쇼핑몰, 가상대학, 가상사무실 및 엔터테인먼트, 에듀테인먼트 등의 다양한 응용분야에 매우 효과적으로 적용될 수 있다. 그림 8은 본 논문을 실제 가상 쇼핑몰 응용분야에 적용시켜 구현한 시스템의 예이다.

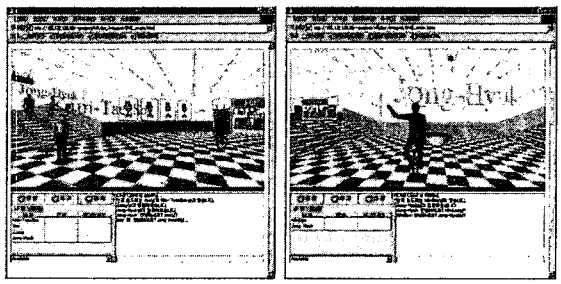

그림 8 가상 쇼핑몰에의 적용

7. 참고 문헌

[1] 시스템공학연구소, 분신의 행동양식처리 및 상호작용 S/W개발, 정보통신부 연구보고서(9P00200-019-822-F), 1997.

[2] Bernhard Jung and Jan-Torsten Milde, An open virtual environment for autonomous agents using VRML and Java, Proc. of VRML 99, pp. 7-11, 1999.

[3] External Authoring Interface Working Group, available at http://www.web3d.org/WorkingGroups/vrml-eai.

[4] Mostafa A. Bassiouni et. al, Performance and Reliability Analysis of Relevance Filtering for Scalable Distributed Interactive Simulation, ACM Transactions on Modeling and Computer Simulation, Vol. 7, No. 3, pp. 293-331, 1997

[5] Funkhouser, T. A., A Client-Server System for Multi-User Virtual Environments, Proc. of Computer Graphics, pp. 85-92, 1997.