공간음향의 실시간 전달을 위한 시스템 구현

김풍혁 *, 오용택 *, 최덕재 **
전남대학교 전산학과 인터넷응용기술연구실
{dcwork, oytzzz}@iat.chonnam.ac.kr, dchoi@chonnam.ac.kr **

Implementation Realtime 3D-Sound Transport System

Pung-Hyeok Kim, Yongtaek Oh, Deokjai Choi
Dept. of Computer Science, Chonnam National University

요 약

네트워크 기반 실시간 멀티미디어 응용 프로그램의 발전이 점점 가속화하는 가운데, 보다 높은 질을 갖는 멀티미디어 데이터에 대한 요구도 점점 커지고 있다. 입체음향 기능은 음향 부분의 가상현실성을 향상시킬 수 있는 방법이며, 공간음향은 입체음향 효과를 구현할 수 있는 소프트웨어적인 인터페이스를 제공해 준다. 본 논문에서는 공간음향 기능을 네트워크 기반 어플리케이션에서 활용하는데 필요한 요소들과 처리 방법에 대해 논의하고 이를 실제로 구현한다. 아울러 구현 과정에서 발견한 문제들에 대해 논의한다.

1. 서 론

입체음향 기술은 사람이 두 귀로 듣는 음향의 도달 시간, 음향의 크기 차이 등의 여러 가지 요소를 재현함으로써 음원의 방향을 인지할 수 있도록 만들어준다. 입체 음향 기술은 다양한 멀티미디어 응용 분야에서 폭넓게 사용되어 왔으며, 스테레오 데이터를 넘어 다채널 오디오를 사용하는 등 더욱 높은 품질을 추구하는 방향으로 발전되어 왔다.

한편 소리 데이터를 사용하는 통신 방식은 인간에게 있어서 가장 기본적인 의사소통 수단이다. 이는 기존 유선 전화 서비스에서부터 VoIP 기술과 다양한 무선 통신 분야에 이르기까지 오랜 시간 동안 다양한 전달 매체상에서 기본적인 통신 수단으로서 자리를 지키고 있다.

본 논문에서는 이러한 음성통신 분야의 실감성을 높이기 위한 방법으로써 공간 음향을 활용하기 위해 필요한 요소들에 대해 논의하고, 기본 개념에 따라 프로그램으로 구현하여 얻은 특성에 대해 설명한다. 이에 따라 2장에서는 프로그램 구현을 위해 고려할 요소들을 먼저 언급한다. 3장에서는 실제 구현 구조와 그 결과에 대해 논의하고, 4장에서는 논문의 결론 및 차후 연구 방향에 대해 기술한다.

2. 배 경

영화와 같이 저장 매체를 기반으로 하는 멀티미디어 데이터는 주로 멀티채널(Multi-Channel) 형태의 오디오 데이터를 사용함으로써 입체음향을 구현한다. 멀티채널 오디오를 기반으로 하는 입체음향 기술에는 AC-3, MPEG-2 오디오, DTS 등이 있다. [1] 하지만 사람은 소리에 의한 공간 지각에 있어서 최종적으로 2개의 귀를 통해서만 이루어지므로, 사람 귀에 전달되는 전송경로를 시뮬레이션하여 입체 신호를 합성하면 그림 1과 같이 스테레오 신호만으로도 입체감을 나타낼 수 있다. [2]

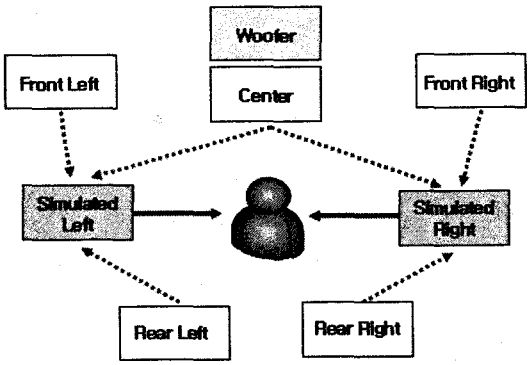

그림 1. 스테레오로 합성된 5.1채널 멀티채널 음향.

따라서 그림2와 같이 네트워크상에서 입체음향 데이터를 전달하여 재생하는 경우에도 이와 같은 접근법을 사용할 수 있다. 이는 특히 방송과 같이 수신자에게 전달할 데이터가 미리 가공된 결과만을 재생하는 경우에는 적절한 구조로 채택할 수 있다. 하지만 이 방법은 실시간성을 요구하는 응용 분야에서는 지연시간을 증가시키는 요소로 작용할 수 있는 문제 외에도, 합성을 통해 얻어낸 결과 데이터가 입체음향 효과를 갖기는 하지만 본질적으로 특정 환경에 대해서만 유효한 결과만을 갖는 한계점을 갖고 있다.

예를 들어 그림 3과 같이 어떤 가상 환경 내에서 음향 측면의 환경 공유 효과를 가져야 하는 경우, 특정 소리에 대해 참여자 A가 듣는 입체 음향 데이터는 참여자 B, C가 들어야 하는 형태와는 다른 재생 결과를 갖는다. 각각의 개체가 그림 상에서 위쪽을 전면으로 향한다고 했을 때, A는 오른쪽 측면, B는 전면에서 소리를 인식해야 한다. 이 상황을 그림 2의 구조 아래에서 해결하기

위해서는 방송 제공자가 참여자 각각에 맞는 스테레오 데이터를 따로 만들어 보내줘야 하므로 가상환경 네트워크 내에 참여자가 많아질수록 불필요한 부하가 가중된다. 결과적으로 그림 2와 같은 구조는 네트워크상에서 동일한 음원 데이터를 사용하면서도 서로 다른 입체감을 가져야 하는 경우에는 적용하기 어려워진다.

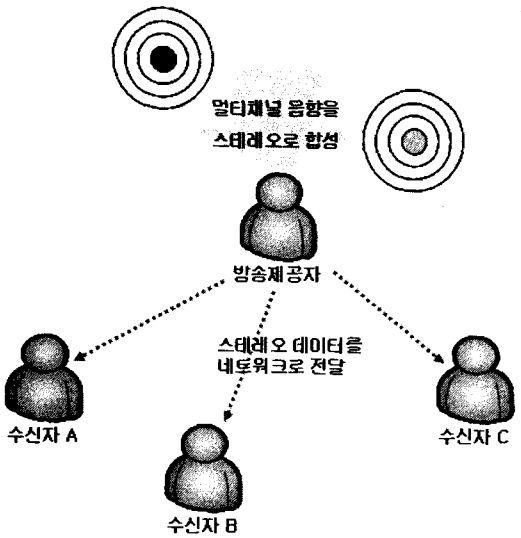

그림 2. 멀티채널을 사용한 입체음향 데이터 전달.

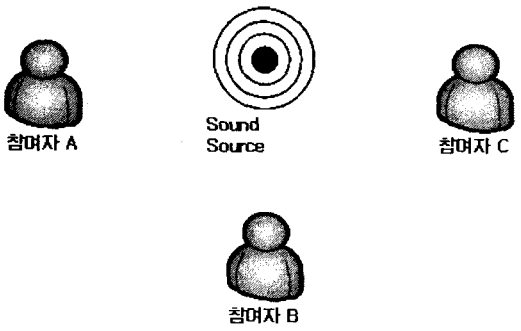

그림 3. 다수의 참여자가 공유하는 가상 환경.

따라서 이와 같은 상황에도 적용할 수 있도록 하기 위해 이전 논문에서 공간음향 기능을 네트워크 기반 어플리케이션에서 활용하는 시스템 설계에 대해 제안하였다. [3] 이 시스템의 기본 출발점은 어플리케이션의 음향 재생 시스템을 공간음향 기능을 활용하고, 전송 시스템에서 사용할 패킷을 속성헤더와 음향 데이터로 구분하는 것, 그리고 속성헤더의 크기를 가변으로 조절할 수 있도록 구성하는 것이다. 본 논문에서는 앞서 제안한 구조를 따르는 간략한 서버와 클라이언트를 구현하고, 그 결과로 얻은 특성에 대해 다룬다.

3. 전송 시스템 구현

PC 상에서 공간 음향 특성을 낼 수 있도록 만들어주는 라이브러리 역시 다양한 종류가 존재한다. 본 논문에서는 음향 시스템으로서 크리에이티브에서 공개하는 OpenAL [4] 라이브러리를 사용하였다. OpenAL 라이브러리는 MS 윈도우와 리눅스를 비롯한 다양한 플랫폼에서 구동할 수 있는 크로스플랫폼 라이브러리이다. OpenAL 이외의 네트워크나 인터페이스 부분은 MS 윈도우의 시스템 호출들을 그대로 사용하였다. 구현 시스템에서는 공간 음향의 개체 속성으로서 위치 속성과 방향 속성 2가지를 사용하였다. 위치는 해당 음향 공간 내에서 그 개체의 위치를, 방향은 개체의 방향을 나타내며 이들 속성은 모두 3차원 벡터로 표현된다. 방향 속성은 청취자 개체에게 있어서 해당 지점 내에서 다시 양 쪽 귀의 위치를 결정하는 요소이다. 따라서 위치가 같더라도 방향이 다르면 공간 내에서 동일한 음원에 대해서도 듣는 소리가 달라질 수 있다.

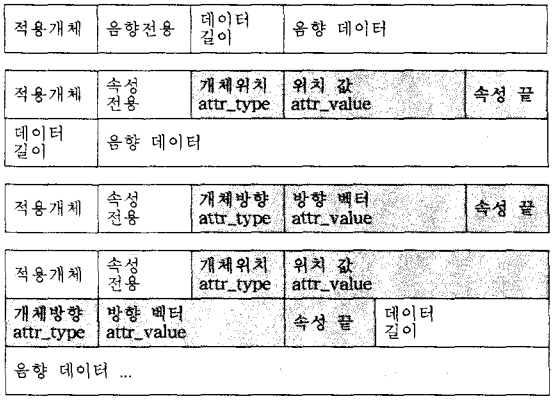

그림 4. 구현된 데이터 패킷 형식.

그림 4는 실제로 구현한 패킷의 형식들을 나타낸다. 이번 구현에서는 전에 제시한 설계와 비교할 때 적용할 개체를 나타내는 부분과 데이터 패킷의 형식을 가리키는 필드의 순서를 바꾸었다. 이는 목표로 하는 성능에 따라 적절한 형태를 찾아야 한다. 그림에서 음영으로 표시된 부분은 해당 패킷에서 음향 데이터가 아닌 속성 헤더를 이루는 부분을 나타낸다. 속성 헤더 내의 값은 속성 종류를 구분하는 구분자와 해당 속성 값이 연달아 붙어서 1쌍을 구성한다. 속성 헤더의 끝에는 속성 헤더의 끝을 표현하는 구분자로 막는다. 이러한 구조를 사용하는 것은 헤더를 구성하는 속성의 개수에 상관없이 헤더를 만들고 해석할 수 있도록 하기 위함이다. 이 경우 속성 개수는 물론이고 순서에도 영향을 받지 않도록 만들 수 있다. 이 경우 만일 모든 속성을 혹은 상당히 많은 속성들을 한 번에 모두 설정해야 하는 경우라면 그만큼 속성 식별자가 차지하는 크기가 증가할 것이므로 오히려 불리할 수 있으나, 그렇지 않은 경우에는 헤더를 구성하는데 있어서 유연성을 얻을 수 있다. 표 1은 이를 처리하기 위한 개념적인 알고리즘을 나타낸 것이다.

```
read attr_type from packet;

while( ATTR_TERMINATE != attr_type )
{
    switch( attr_type )
    {
        case ATTR_POSITION:
            read attr_value from packet;
            set attr_value as position;
        break;

        case ATTR_DIRECTION:
            read attr_value from packet;
            set attr_value as direction;
        break;
    }

    read attr_type from packet;
    code = attr_type;
}
```

표 1. 속성헤더 해석 및 처리.

구현한 어플리케이션에서 서버는 공간 내 개체들의 정보를 관리하는 한편 이에 맞는 음향 데이터를 제공하고, 클라이언트는 이를 수신하여 해석하고 재생하는 역할을 한다. 그림 5는 서버 프로그램의 인터페이스 가운데 좌표 할당기를 나타낸다. node 0은 서버 자신을 표현하는 기호이며, node 1은 접속해 있는 클라이언트를 나타낸다. 밑에 표시되는 값은 각 개체들의 위치값과 방향 벡터이다.

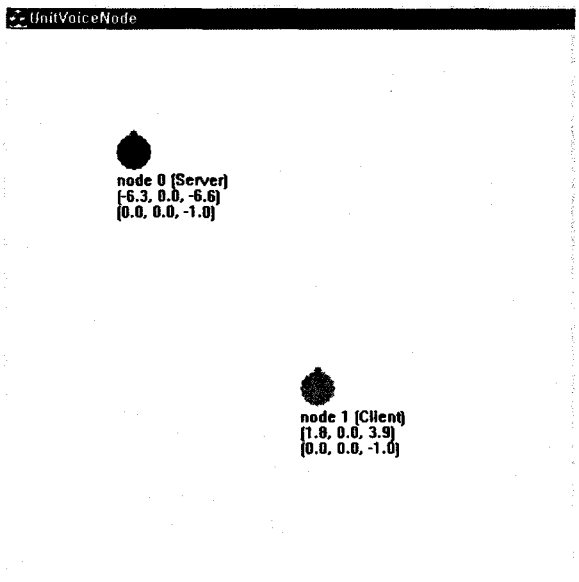

그림 5. 서버 인터페이스의 좌표 할당기.

그림 5에서 두 개체 모두 방향은 위쪽이 전면으로 설정되어 있다. 따라서 이 상태에서 클라이언트는 서버가 보내는 음향 데이터를 재생할 때 왼쪽 전면에서 오는 것으로 인식하게 된다. 이런 위상은 서버에서 조작하는 것에 따라 실시간으로 변화한다. 그림 6은 서버에서 보내는 데이터를 수신하는 클라이언트를 나타낸 것이다.

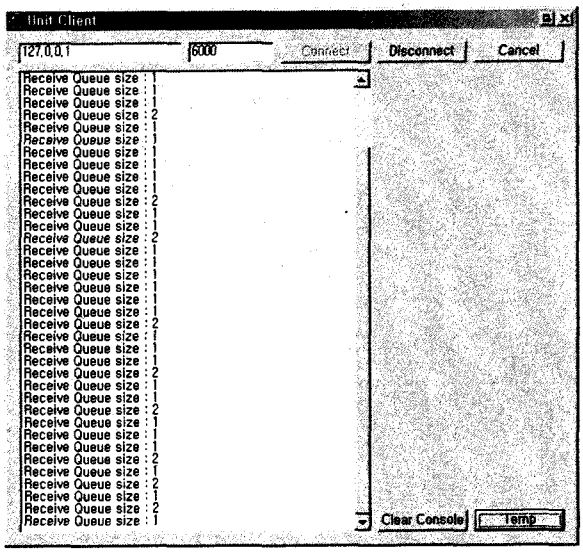

그림 6. 클라이언트 인터페이스.

주목할 점을 2가지 찾아볼 수 있는데 하나는 서버에서 개체를 관리하고 정보를 할당하는 문제이다. 서버는 서버 자신이 음향 데이터를 제공하는 것이므로 자신의 위치가 변화해도 자신에게 들리는 소리가 변하도록 조작하면 안 된다. 하지만 node 1 의 위치나 방향을 조작하는 경우 이를 처리할 수 있는 명령 패킷을 발생시켜서 보내줘야 한다. 반대로 클라이언트 프로그램 측면에서는 node 1이 자신을 표현하는 개체가 되고, node 0은 자신의 외부 개체가 된다. 클라이언트는 데이터를 받아서 재생만 하면 되기는 하지만 공간음향 연산을 다른 개체에 적용하는 상황을 고려하는 것이 필요하다. 여기에 관해서는 해당 어플리케이션이 목표로 하는 상황에 맞게 적절한 처리기 및 전송기를 작성해야 한다.

다른 하나는 서버에서 발생하는 변화 이벤트를 속성 헤더로 변화시키는 과정이다. 그림 5에서 구현한 서버 좌표 할당기의 경우 윈도우 프로그램의 드래그 이벤트를 좌표 할당에, 키보드 이벤트를 방향 할당에 사용하고 있는데, 발생하는 이벤트들을 그냥 순차적으로 속성 헤더로 사용할 경우 너무 많은 패킷들이 발생하여 이벤트 큐가 적체되는 현상을 보일 수 있다. 이는 클라이언트에서 재생할 때 속성 변화를 반영하는 것이 느려지는 결과를 가져오게 되므로 실시간 데이터를 사용해야 하는 응용 분야에서는 부적절한 결과를 가져올 수 있다. 따라서 이 역시 이벤트 변화를 적절히 반영할 수 있도록 이벤트 큐를 구성해야 한다.

4. 결론

본 논문에서는 앞서 제시한 실시간 전달 시스템의 설계를 실제로 구현하기 위해 고려해야 하는 점들을 논의하고 이를 어플리케이션으로 구현하는 한편, 구현 과정에서 나타난 특이점들을 논의하였다. 제시한 시스템은 주로 네트워크 기반 회의 혹은 의사소통 환경에서 가상 현실성을 높일 수 있는 기능을 의도하고 있다. 본 논문에서는 RTP 와 같은 패킷 네트워크상에서의 전통적인 문제에 대해서는 다루지 않고, 음향 데이터에 공간 정보를 부가적으로 실어 보내는 방법에 대해서만 집중하였다.

향후 연구과제로서는 제시한 전송 프로그램의 수용성과 기능성을 확대하는 문제를 다룰 예정이며, 전통적인 멀티미디어 스트리밍 문제를 포함하여 통합된 시스템으로 발전시킬 예정이다.

참고문헌

[1] 임충규. 온라인 3D 게임 기술. 한국정보처리학회 학회지. 2003년 1월. 제10권 제1호. 82-87쪽.

[2] 류승문. Ubiquitous 공간에서의 무선 디지털 오디오 시장. 한국통신학회지. Vol.22, No.10 October. 2005. 46-57쪽

[3] 김풍혁. 공간음향의 실시간 전달을 위한 시스템 설계. 2006 한국컴퓨터종합학술대회 논문집. Vol.33, No.1. p.121-123.

[4] OpenAL 1.1 Specification. http://openal.org