

그래픽 콘텐츠 개발자를 위한 입체음 재생 인터페이스 기본 설계 및 구현

원용태*, 장봉석**, 안동순***, 곽훈성****

요약

입체음 재생을 위한 하드웨어 및 소프트웨어 기술발전에 의해 입체음과 입체영상을 적용한 3D 가상 공간은 보다 증강된 입체감 및 생동감을 제공할 수 있다. 그러나 영세한 지역 영상콘텐츠 전문제작자들과 일반인들이 입체음을 구현하기 위해서는 고가의 입체음향 엔진구입, 전문 인력 투입 및 고급 프로그래밍을 작성해야 하는 어려움이 있다. 그러므로 입체음 기술의 저비용 활용을 위해서 단순하면서도 효율적인 입체음 재생 인터페이스 개발이 필요하다. 이를 활용하게 되면 영상 제작전문가들이 손쉽게 입체음 재생기술을 콘텐츠 제작물에 추가할 수 있다.

본 논문은 입체음 재생 인터페이스 개발을 위한 기초연구 단계로서 다음 내용을 설계하고 구현한다. 먼저 PC 기반 시스템에서 모노음을 입체음향으로 재생하는 주요한 모듈을 구현한다. 두 번째로 3D 영상과 입체음을 연동하는 모듈을 개발한다. 본 논문에서 구현된 모듈은 청취자가 가상공간 상에서 이동함에 따라 음원의 위치 및 서라운드 효과를 인지할 수 있도록 한다. 추후 본 연구는 사운드와 이동객체의 동기화 및 HRTF 구현을 추진하여 보다 완벽한 입체음향 재생 인터페이스를 완성 할 예정이다.

Preliminary Design and Implementation of 3D Sound Play Interface for Graphic Contents Developer

Yong-Tae Won*, Bongseog Jang**, Dong-Soon Ahn***, Hoon-Sung Kwak****

Abstract

Due to the advance of H/W and S/W techniques to play 3D sound, the virtual space contented by 3D graphics and sounds can provide users more improved realities and vividness. However for the small 3D contents developers and companies, it is hard to implement 3D sound techniques because the implementation requires expensive sound engines, 3D sound technical understanding and 3D sound programming skills. Therefore 3D-sound-playing-interface is necessary to easy and cost-effective 3D sound implementation. Using this interface, graphics experts can easily add 3D sound techniques to their applications.

In this paper, the followings are designed and implemented as a preliminary stage in the way of developing the 3D sound playing interface. First, we develop 3D sound S/W modules converting mono to 3D sound in PC based systems. Second, we develop the interconnection modules to map 3D graphic objects and sound sources. The developed modules in this paper can allow the user to percept sound source position and surround effect at the moving positions in the virtual world. In the coming works, we are going to develop the more completed 3D sound playing interface consisted of the synchronization technique for sound and moving objects, and HRTF.

Keyword : 3D Sound, Sound Play Interface, 3D Sound and Graphic Mapping, Contents Developer

1. 서론

※ 제일저자(First Author) : 원용태
접수일자:2008년06월03일, 심사완료:2008년06월18일
* 전북대학교 영상공학과, aniwon@empal.com
** 목포대학교 정보공학부(교신저자)
** 목포대학교 정보공학부
**** 전북대학교 영상공학과

입체음향(3D 사운드)기술은 사운드에 입체감을 부여하는 기술로서 입체 가상공간 상에 적용하게 되면, 입체 가상공간의 청취자와 음원의 위치에 따라서 청취자가 음원의 위치, 실제음 재생

효과 및 주위 환경을 인식할 수 있는 가상음을 생성할 수 있도록 하는 기술이다. 특히 최근에는 서라운드 시스템의 멀티채널 방식을 적용하지 않더라도 일반 PC 스테레오 시스템 상에서도 입체음을 재생할 수 있도록 개발되고 있다.

입체음향기술은 기본적으로 HRTF (Head Related Transfer Function, 머리전달함수)를 적용하여 위상정위를 구현된다. 위상정위는 청취되는 음원의 위치가 전후좌우상하 위치 및 거리감을 인식할 수 있도록 한다[1]. 머리전달함수는 실험실에서 더미헤드 주위에 배치된 여러 위치의 음원에서 발생한 사운드를 수집한 후에 사운드 필터링 함수를 구하는 것이다. 위상정위에 더하여 음장제어기술을 적용한다. 음장제어는 공간감을 인지할 수 있도록 하는 사운드 필터를 연구하는 분야이다. 가상공간의 재질과 공기의 밀도 그리고 음의 전달지연에 의해서 이와 같은 공간감을 인식할 수 있도록 할 수 있다[2][6].

이러한 입체음향기술은 음원의 위치 및 거리감을 인식하여 입체음향을 구현하는데 어려운 점이 있으나 본 논문의 연구에 특징은 입체가상공간을 구축하는데 있어서, 손쉽게 기존의 3D 그래픽 기술에 입체음 기술을 추가할 수 있는 입체음 생성기술을 개발하여 가상체험관, 온라인 게임 등의 개발자들이 입체음 기술 적용의 어려움을 해소할 수 있는 인터페이스를 개발하고자 한다. 즉, 입체음향 구현의 어려움을 입체음 재생 인터페이스를 제공하여 쉽게 개발에 활용 할 수 있는 기술을 제공하고자한다. 3D 그래픽스 기술은 이미 많은 응용 콘텐츠제작에서 활용되고 있지만, 거의 모든 응용 프로그램은 3D 그래픽스 기술의 입체감 증진에 대한 내용 위주로 개발이 되어 왔다[2]. 그러나 최근에 입체음향 재생 기술의 적용에 대한 요구가 증대되고 있으며 실제 입체음향기술을 적용하였을 때, 가상공간 상에서 보다 증진된 입체감 및 생동감을 인식할 수 있다. 특히 게임 분야에서 고가의 게임 엔진은 입체음향기술을 적용한 사운드 엔진이 출시되고 있으며 이를 활용하여 보다 입체감이 우수한 게임이 출시되고 있다. 그러나 이러한 게임엔진은 고가이며 이를 구입하여 응용 프로그램을 제작하기 위해서는 영세한 콘텐츠 제작 기업들에게는 경제적인 문제가 있다.

본 연구는 입체음향 구현 인터페이스를 개발

하는 장기적인 목표로 추진되고 있으며 본 논문에서는 그 기초단계로서 지금까지 설계 구현된 내용을 소개한다. 먼저 PC 기반의 스테레오 시스템에 구현된 입체가상공간의 청취자와 음원의 위치에 따라 서라운드 및 음원위치 파악 모듈을 구현하였다. 그리고 영상객체와 음원의 연동모듈을 설계 구현하였다. 구현된 내용은 실제 청취자를 통한 사운드 인지정도를 조사하여 성능을 평가하였다. 향후 본 연구는 이동영상객체 및 머리 전달함수를 구현하도록 추진될 예정이다[7][8].

본 논문의 구성은 다음과 같다. 2장에서는 입체음 재생 모듈 설계에 대해서 설명하고, 3장에서는 구현된 기술의 구현과 실험결과를 소개한다. 마지막으로 4장에서는 결론과 향후 계획에 대해서 소개한다.

2. 입체음 재생 모듈 설계

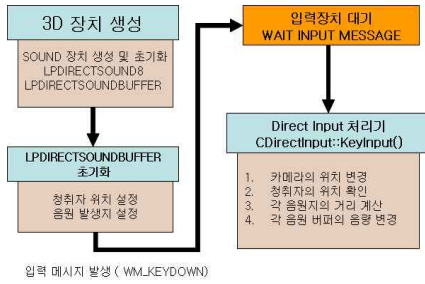
2.1 입체음 재생 환경

PC 기반 스테레오 환경에서 입체음 재생시스템은 PC 게임 및 웹3D 가상공간과 같은 어플리케이션에서 입체음 재생을 적용하고자 하는 경우 이용 가능하다.

입체음 재생 시스템 개발을 위한 구현 플랫폼 및 모듈은 다음과 같은 내용을 고려하였다.

입체음 기술 내용을 응용하여 DirectX 기반의 DirectSound3D API를 활용한 입체음 재생시스템을 개발하여 고가의 음향장비 사용 없이 스테레오 재생시스템에서 헤드폰을 이용하여 가상현실 공간의 현장감 및 입체감을 증폭하는 기술을 구현한다.

모노음을 음원으로 설정하고 가상공간 상에 다수 위치하게 하며, 스테레오 시스템에서 모노음원으로부터 입체음 생성을 가능하게 하는 기술을 구현한다. (그림 1)은 3D 가상공간 좌표 상에서 청취자가 이동하면 모노음원 발생지로부터 청취자의 위치에 따라 입체음이 발생하는 과정에 대한 상위설계도를 보여준다.



(그림 1) 청취자의 위치에 따른 입체음 재생 과정

2.2 입체음 구현 모듈

입체음 실행환경은 DirectX SDK 9.0C 버전 설치 후 DirectX API를 적용하여 가상 입체음을 재생하며 입체 가상공간 그래픽 구축과 독립적으로 입체음 재생이 가능하도록 개발한다. 입체음 재생 모듈에 포함된 기술적 구현 내용은 다음과 같다.

도플러 효과 구현은 청취자의 이동 속도에 따라 사운드 효과를 재생한다.

$$f' = f \cdot [(v \pm v_L) / v] \tag{1}$$

식 (1)에서 f' 는 도플러 효과 주파수, f 는 원음의 주파수, v 는 음의 속도이며, v_L 은 청취자와 음원의 이동 속도이다. v_L 값을 알면, 도플러 효과 f' 를 구할 수 있다. 도플러 효과는 청취자에게 접근하는 사운드는 고음(주파수가 커지고)이 되고 멀어지는 사운드는 저음(주파수가 낮아지는)이 되어가는 효과를 표현할 수 있다 [3][7][8].

거리에 따른 사운드 볼륨 제어는 거리에 따르는 사운드 감쇄 값을 계산한다. 최소 사운드 청취 거리 값과 그 최소 거리에서 청취되는 최대 볼륨 값을 이용한다. 사운드 감쇄값은 음원의 위치에서 멀어질수록 로그함수에 의한 사운드 볼륨의 감쇄효과(음압의 저하 현상)를 나타낸다. 가상공간 상의 사운드 발생 객체에 따른 최대 최소 볼륨 조정은 가상 입체공간에 존재하는 사운드 발생 객체의 발생음 최대 최소 크기의 설정에 따른 다양한 볼륨을 구성한다. 예를 들어서, 폭포 소리와 새소리는 그 최대 및 최소 사운드 크기(음압의 차이)가 다름을 표현하여야 된다.

사운드 방향 효과 부여에서는 사운드 방향은 사운드콘으로 표현되고, 내부 콘 각도와 외부 콘 각도로 나누어서 사운드 크기가 조절 가능함으로 사운드의 청취 범위를 사운드 콘의 형태를 구성하여 객체 별로 지정하게 구현 가능하다.

머리전달함수는 입체적 위치감 인지필터 값을 DB에 저장 한 후에 실시간 처리가 필요하다. 실제 DirectX에는 사운드 재생을 CPU 처리로 인하여 머리전달함수의 완벽한 구현이 현실적으로 어려운 부분이다[4][5]. 머리전달함수를 완벽하게 구현 가능하도록 하기 위해서는 사운드카드에 머리전달함수를 내장한 음원 칩과 드라이버, 그리고 그것을 지원해 주는 버전의 DirectSound가 필요하다. 그러므로 본 모듈에서는 극히 일부 제한된 범위에서 적용 가능하도록 구현한다.

입체음 재생 처리 중에 하나인 그래픽 객체의 이동 거리 변환 기능을 위한 음원에서 청취자와의 거리는 DirectX 상에서는 미터 단위로 정의되지만 입체 그래픽 환경에서의 이동단위는 픽셀단위로 정의 된다. 그러므로 두 가지 정의를 매핑 시키고 변환하는 모듈을 설계 구현한다.

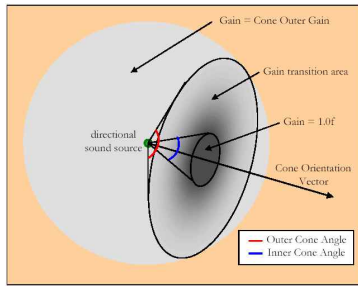
사운드 모드 설정은 사운드 버퍼에 사운드 모드를 설정하도록 하여서 청취자와 사운드 소스의 구현을 응용 프로그램에 맞게 구성할 수 있다. 입체음 버퍼를 설정할 때, 변수를 DS3DMODE_HEADRELATIVE로 설정하면 청취자의 위치가 변하는 경우 입체 가상공간 상에서 일인칭 시점인 경우에 적합하며, DS3DMODE_NORMAL인 경우는 청취자의 위치가 변하지 않는 경우에 적합한 설정이다. DS3DMODE_NORMAL인 경우는 디폴트로 설정하도록 되어있다.

그리고 DS3DMODE_DISABLE은 입체음 효과를 제거하는 경우이며 저사양의 PC에서 필요한 설정이다. 음원은 사운드가 가상공간상에 전이되는 방향과 범위를 나타낼 수 있으며 작은 원추가 내부 콘이며 큰 원추가 외부 콘이 되고, 각각의 콘은 각도를 가지고 있다. 또한 음원에 대해서 콘 및 콘 외부 지역에 대한 획득 값이 할당 된다. 음원과 내부 콘의 평면은 gain=1로 가장 높게 할당되며, 그 외 평면은 gain=1로부터 점차적으로 작은 값이 할당된다. 획득 값이 1인 경우가 가장 명확하고 큰 사운드를 듣는 지역이 된다.

청취자 위치에 따라서 만약 청취자가 내부 콘

에 위치한 경우는 사운드는 균일하게 출력이 되며 내부 콘과 외부 콘 사이에 위치 시 거리에 따라서 일정한 크기로 사운드가 감소된다. 그리고 외부 콘의 바깥쪽에 청취자가 위치하면, 사운드가 발생하지 않도록 혹은 정해진 dB 만큼 감소 모델링 하도록 한다.

DirectSound3D에서는 콘의 방향을 결정하는 X, Y, Z 방향변수와 내부 콘 각도, 외부 콘 각도, 그리고 내부 콘과 외부 콘 사이의 볼륨 백터를 증강 변수로 입력 받을 수 있다. 이러한 지정 가능한 값들을 라이브러리 함수를 이용하여 지정할 수 있도록 개발되었다.



(그림 2) 사운드 방향과 사운드 cone

또한 입체음 재생기술 이외로 입력장치 및 유저 입력 제어 방법 그리고 이와 관련된 사운드 제어 구현 방법에 대한 내용은 다음절에서 거론한다.

2.3 외부입력에 의한 제어

2.3.1 입체공간 상의 입력장치 및 유저 입력 제어

Direct Input은 DirectX에 포함되어 있는 응용 프로그램 인터페이스 API로 게임용 조이스틱, 마우스, 키보드와 같은 장치를 제어하기 위한 라이브러리이다. 본 논문의 연구에서는 Win32 API을 이용하여 윈도우 메시지를 처리한 후, 입력과 관련된 메시지를 Direct Input을 통해 처리하는 방법을 사용한다.

- ① WIN32 메시지 처리 함수: 윈도우 이벤트 기반 메시지를 Direct Input 객체함수에 넘겨주는 처리를 담당한다.

- ② Direct Input 장치 생성: 응용프로그램으로부터 입력 장치 메시지를 받아 처리하기 위해서는 모든 입력장치를 제어하기 위한 LPDIRECTINPUT8 객체를 생성하여야 하고, 그리고 각 장치 요소를 제어하기 위한 LPDIRECTINPUT DEVICES 객체를 생성하여야 한다. Direct Input 관리 객체와 키보드를 위한 장치 객체를 관리하기 위한 클래스를 구현한다.

- ③ 키보드 처리 루틴: CDirectInput 객체에 생성된 LPDIRECTINPUT DEVICES 변수는 응용프로그램으로부터 메시지를 전달 받아 입력 값에 따라 카메라의 위치의 변경과 같은 변환들을 처리하고 있다. 키보드에서 UP, DOWN, LEFT, RIGHT 키에 의해 화면의 방향과 거리에 변환을 주는 내용을 구현한다.

2.3.2 외부 입력에 의한 입체음 제어

청취자의 위치에 따라 가상 입체공간 좌표 상에 존재하는 음원 발생지에서 발생하는 사운드들이 음량의 크기가 다르게 발생하도록 하여야 한다. 이를 구현하기 위해서는 사운드 장치를 관리하는 LPDIRECTSOUND8 와 각각의 사운드파일을 제어할 수 있는 LPDIRECT SOUND DBUFFER 객체가 생성되도록 구현한다.

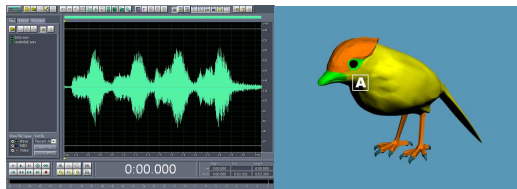
- ① 사운드 장치 객체: LPDIRECTSOUND8 객체는 SOUND 장치를 제어하는 객체이다. 사운드 장치를 효율적으로 관리하기 위한 객체 클래스를 생성하여 구현한다.
- ② 음원 객체: 각 사운드파일을 사운드 버퍼에 저장하고 제어하기 위한 LPDIRECTSOUNDBUFFER를 효율적으로 처리할 수 있는 객체 클래스 내용과 음원 생성을 구현한다.
- ③ 윈도우 활성화 시 사운드 작동: 응용프로그램이 윈도우 화면에서 활성화 되었을 때 작동될 수 있도록 메시지 처리부분에서 처리하도록 구현한다.
- ④ 입력장치의 변환에 따른 사운드 제어: 키보드와 같은 입력장치에 의해 청취자의 위치가 바뀌게 되면 CDirectInput 객체는 KeyInput함수에서 CD- Sound::SetChangePos()을 호출하여 각 소리의 음량을 제어할 수 있도록 구현한다.

3. 입체 가상공간에서 입체음 구현과 실험결과

3.1 사운드 음원 설정 및 입체 가상공간 구현

입체 가상공간을 구현하기 위한 시나리오는 새와 폭포는 고정된 음원이고 청취자는 키보드를 이용하여 이동 가능하며 이동속도는 일정하다. 카메라 위치 혹은 시점은 y축 이동은 없고, x축과 z축으로 이동이 가능하다. 사운드는 DirectSound3D사운드 제어 라이브러리와 추가적으로 구현된 모듈을 이용한다.

(그림 3)은 가상공간 상에 제작된 입체 버드이며, 음원은 모노음이며 재생시간은 3.3초이고 4번의 반복 음을 들려준다. [A]는 입체 버드 그래픽에 위치하는 모노음원의 위치를 보여준다. 만일 버드의 날개소리를 음원으로 제작하면, 음원의 위치를 버드의 날개부위에 위치할 수 있다.



(그림 3) 버드의 사운드파형과 음원위치 [A]

(그림 4)는 폭포에 대한 사운드 파형으로 음의 재생시간은 5.3초이고 폭포수의 사운드를 들려준다. [B]는 폭포 그래픽에 위치하는 음원의 위치를 보여준다. 마찬가지로 음원위치는 변경 가능하다.



(그림 4) 폭포의 사운드파형과 음원위치 [B]

3.2 입체음 재생 모듈 구현

본 모듈 구현은 DirectX 9.x SDK 버전을 이

용하였다. 3차원 게임과 같은 가상 공간상에서 입체음향 효과를 얻고자 한다면 본 라이브러리를 이용하면 원하는 결과를 쉽게 얻을 수 있다. C++언어로 구현되었으며, 개발 통합 환경은 비주얼 스튜디오 2003.NET을 이용하였다. 기존의 라이브러리를 사용하기 위해서는 DirectX SDK 9.0C 버전을 다운로드 받아 설치하여야 한다.

입체음 재생 모듈을 제공하는 파일들은 헤더 파일 2개(S3DSystem.h, S3DControl.h)와 라이브러리 파일 1개(S3DL.lib)로 구현하였다. 본 프로그램 사용자는 파일들을 프로젝트에 추가하여 함께 컴파일을 하여야 한다.

S3DControl.h은 입체 가상공간에서 사용하는 픽셀단위와 사운드 공간에서 사용하는 미터 단위를 매핑하여 서로 변환하는 함수에 대한 정의가 되어있다. S3DSystem.h 파일은 입체음 재생에 활용하기 위한 모든 인터페이스 함수들이 정의되어져 있다.

S3DL.lib 라이브러리는 음원생성과 청취자 설정 그리고 입체음 재생을 위한 파일들로 구성된 라이브러리이다. 그 중에서 시스템 초기화, 음원 설정 그리고 청취자 설정에 관한 소스코드를 다음 (그림 5)(그림 6)(그림 7)에서 보여준다.

```

// Sound 장치 가동
m_sndSystem = new CS3DSystem();

// 청취자
////////////////////////////////////
if (FAILED(m_sndSystem->Initialise(m_hWnd) ) )
    return E_FAIL;

// [방법1] 함수를 이용하는 방법
m_sndSystem->SetListenerPosition(
S3DConvertPixelToMeter(m_Actor.x,m_nPixelPerOneMeter),
S3DConvertPixelToMeter(m_Actor.y,m_nPixelPerOneMeter),
S3DConvertPixelToMeter(m_Actor.z,m_nPixelPerOneMeter));
    
```

```

// 청취자의 정보를 설정
m_sndSystem->SetListenerVelocity(5, 1, 3); // 청취자의 정보를 설정
m_sndSystem->SetListenerDopplerFactor(0.2f);
m_sndSystem->SetListenerRolloffFactor(0.8f);
m_sndSystem->CommitListenerParameters();

// [방법2] 사운드 음원 등록처럼 파라미터 구조체(LPDS3DLISTENER)를 이용하는 방법도 있다.
LPDS3DLISTENER params;
m_sndSystem->GetListenerParameters(params);
params.vVelocity 값을 적용
m_sndSystem->CommitListenerParameters();
    
```

(그림 5) 사운드 시스템 생성과 청취자 설정 소스코드

```

// 새소리 등록
// FLAG : DSBCAPS_CTRL3D
DSBCAPS_MUTE3DATMAXDISTANCE
// MODE : DS3DMODE_NORMAL
DS3DMODE_HEADRELATIVE
(
GUID_NULL,DS3DALG_NO_VIRTUALIZATION,DS3DALG_HRTF_FULL, DS3DALG_HRTF_LIGHT
)
int idBird;

if(m_sndSystem->AddWav("/sounds/bird.wav", &idBird,
DSBCAPS_CTRL3D
DSBCAPS_MUTE3DATMAXDISTANCE,
DS3DALG_HRTF_FULL)
return FALSE;

LPDS3DBUFFER ldsSBBird; // 파라미터를 이용한 설정방법
ldsSBBird =
m_sndSystem->GetSoundParameters(idBird); // 파라미터를 가져온다
ldsSBBird->dwMode = DS3DMODE_NORMAL;

D3DVECTOR vPosition;
vPosition.x =
S3DConvertPixelToMeter(150,m_nPixelPerOneMeter);
vPosition.y =
S3DConvertPixelToMeter(2,m_nPixelPerOneMeter);
vPosition.z =
S3DConvertPixelToMeter(-450,m_nPixelPerOneMeter);
ldsSBBird->vPosition = vPosition;
    
```

```

D3DVECTOR vVelocity;
vVelocity.x = 3.0f;
vVelocity.y = 2.0f;
vVelocity.z = 3.0f;
ldsSBBird->vVelocity = vVelocity;

ldsSBBird->flMinDistance = 1.0f;
ldsSBBird->flMaxDistance = 35.0f;

m_sndSystem->CommitSoundParameters(idBird);

m_sndSystem->PlaySound(idBird,TRUE); // 소리를 작동시킨다.
    
```

(그림 6) 버드 사운드 음원 설정 소스코드

```

// 폭포소리 등록
// FLAG : DSBCAPS_CTRL3D
DSBCAPS_MUTE3DATMAXDISTANCE
//GUID
GUID_NULL,DS3DALG_NO_VIRTUALIZATION,DS3DALG_HRTF_FULL, DS3DALG_HRTF_LIGHT
// MODE : DS3DMODE_NORMAL
DS3DMODE_HEADRELATIVE
(
)
int idFall;

if(!m_sndSystem->AddWav("/sounds/waterfall.wav", &idFall,
DSBCAPS_CTRL3D
DSBCAPS_MUTE3DATMAXDISTANCE,
DS3DALG_HRTF_FULL) )
return FALSE;

m_sndSystem->SetSoundPosition(idFall,
S3DConvertPixelToMeter(0,m_nPixelPerOneMeter),
S3DConvertPixelToMeter(25,m_nPixelPerOneMeter),
S3DConvertPixelToMeter(-10,m_nPixelPerOneMeter));

m_sndSystem->SetSoundVelocity(idFall, 3.0f, 1.0f, 3.0f);

m_sndSystem->SetSoundMode(idFall,DS3DMODE_NORMAL);
    
```

```
m_sndSystem->SetSoundMinDistance(idFall, 15.0f);
m_sndSystem->SetSoundMaxDistance(idFall, 300.0f);
m_sndSystem->PlaySound(idFall,TRUE); // 소리를 작
동시킨다.
```

(그림 7) 폭포소리 사운드 음원 설정 소스코드

3.3 입체 가상공간에서 입체음 실험결과

입체 가상공간의 전체적인 구조는 다음과 같다. 가상공간의 중앙에는 강물이 흐르고 정면의 바위로부터 폭포수가 내려온다. 강물의 좌우측은 잔디로 꾸며져 있으며 새 한 마리가 앉아 있는 공간으로 표현되어 있다. 그리고 화면의 윗부분에는 청취자, 폭포음원, 버드음원의 현 위치가 표현되며 px 는 픽셀을 표시하며 m 은 미터로 두 가지 방법에 의해서 현 위치를 표현하도록 되어있다.

구현된 입체 가상공간 상에는 두개의 음원이 존재한다(버드소리 와 폭포소리). 그리고 청취자는 키보드 (UP, DOWN, LEFT, RIGHT)를 이용하여 이동한다. 버드소리는 짧고 날카롭게 반복해서 발생하며 폭포소리는 크며 우렁차게 들리도록 되어 있다. 버드소리와 폭포소리는 미리 녹음된 소리를 모노 웨이브파일로 저장하였으며 수초 정도의 길이로 녹음되었고, 이를 반복해서 재생하게 되며, 청취자의 위치와 두 음원에 대해서 지정한 사운드 재생 관련 변수 및 함수에 따라서 입체음을 지속적으로 재생할 수 있다. 사운드 음원은 두 경우다 모노음이며 8bit 샘플링 그리고 해상도 8비트를 적용한 단순 모노 저해상도 사운드이다.

(그림 8)에서 폭포의 음원위치는 [0, 2, -1]에 위치하고 있으며 버드의 음원의 위치는 [15, 0, -45]의 위치에 위치함을 알 수 있다. 청취자의 위치가 버드 가까이에 있을 때는 버드의 소리가 크게 들리고 폭포 가까이 이동해가면 버드의 소리는 점차 줄어들고 폭포수의 소리가 크게 들린다. 이동에 따른 음원의 고저는 물론, 위치(동서남북)에 따른 음원의 변화를 확인할 수 있다. 또한, 청취자가 폭포 정면에서 목조다리로 이동할 때 사운드는 정후면 및 좌우 위치에 대한 음원 효과를 느낄 수 있다.



(그림 8) 입체 가상공간 실행화면 및 음원과 청취자 위치에 대한 입체공간상의 좌표값

본 연구의 논문에서 구현한 내용을 헤드폰을 착용한 30명의 사용자에게 실행 하도록 한 후에 그 결과를 다음의 <표 1>로 요약하였다. 버드 및 폭포 소리에 의한 청취자의 위치파악은 30명 모두 인지하여 설계된 모듈이 공간 거리감을 대략적으로 정확하게 인지할 수 있다는 것을 알 수 있다. 입체 가상공간 상에서 이동 속도는 일정하게 구현되었기 때문에 도플러효과를 30명 전원이 인지하지 못하였으며 이 부분은 향후 보완되어야 할 사항이다. 마지막으로 폭포수 앞 중앙에서 청취자가 360도 회전을 하면서 전후좌우 사운드 인지에 대해서는 대략 20%정도가 후방 인지를 하였다.

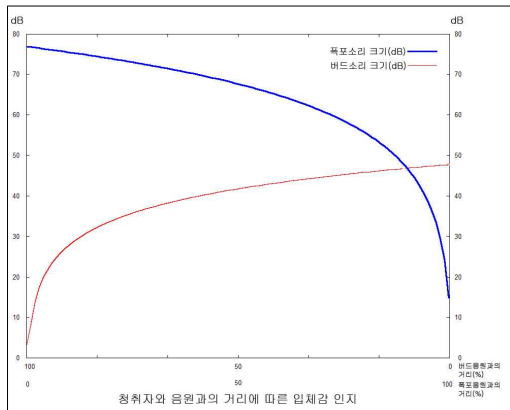
<표 1> 입체감 인지 결과의 통계

수행내용	입체감 인지	불인지
폭포소리에 대한 위치감	30	0
속도에 의한 위치감	0	30
버드소리에 대한 위치감	30	0
음원 후방위치 감지	5	25

(그림 9)는 청취자와 음원과의 거리에 따른 음원의 크기와 입체감의 인지를 나타낸다. 청취자가 폭포 앞에 있을 때 폭포소리는 최대가 되고 버드 쪽으로 이동하게 되면 폭포소리는 약해지며 버드소리가 강해진다. 여기서 폭포소리의 크기는 버드소리보다 크기 때문에 약한 소리를

계속들을 수 있다. 반대로 버드 앞에서 폭포로 이동 할 경우 버드소리는 약해져서 폭포 근처에서 음원은 들리지 않게 된다. 이번 실험 결과 청취자가 가상공간 상에서 이동함에 따라 음원의 위치 및 서라운드 효과를 인지하는 것으로 나타났으며 향후 HRTF를 적용하게 되면 보다 정확한 음상정위가 가능하리라 사료된다.

결론적으로 본 연구의 논문에서 구현한 내용이 사운드 입체감 제공은 가능하나 보다 미세한 입체음향 구현을 위해서는 머리전달함수 및 음장제어기술등과 같은 다양한 모듈의 추가적인 구현이 필요함을 알 수 있다.



(그림 9) 청취자와 음원과의 거리에 따른 입체감 인지 결과

4. 결론 및 향후 과제

입체 가상공간에서 입체음 재생 실험결과를 각각의 기능별로 정리하면 첫째, 도플러 효과는 청취자가 음원에 접근하면 주파수가 높아지고 음원으로부터 멀어지면 주파수가 낮아진다. 그러나 이동속도가 균일하게 구현됨으로서 정교한 표현은 다소 무리가 있었다. 둘째, 감쇄효과는 음원의 설정에 의해서 청취자가 버드에서 폭포수 위치로 이동하는 경우 버드소리는 감쇄효과가 확실하게 작아지고 폭포소리는 더 커진다. 음원의 최대 최소 볼륨 지정이 가능하며 버드와 폭포소리의 강도에 대한 제어가 가능하다. 셋째, 사운드 콘 설정은 콘의 범위를 벗어난 지역에서는 무음 상황이 인식되도록 제어가 가능하다. 넷째, 머리전달함수 적용은 DirectSound3D API 만

으로는 완전한 적용이 어렵다. 향후 하드웨어적인 사운드 개발 인터페이스 활용과 개발이 필요하다. 다섯째, 사운드와 공간 이동거리 변환 기능은 사운드 1m 당 10픽셀로 적용하였으며 사운드 1m 당 임의의 픽셀로 지정하여 테스트가 가능한 인터페이스를 추가할 것이다.

추후연구로는 우선적으로 HRTF를 적용하는 모듈 그리고 그래픽객체의 이동속도를 설정하는 알고리즘과 인터페이스를 개발한다. 둘째, 음상정위가 포함된 음원-그래픽객체 동기화 기술 개발로 가상공간을 이동하는 모델링 객체로부터 수신된 메시지를 통해서 언제 어디서 어떤 사운드가 플레이되는지 제어하는 기술과 음장제어기술을 적용하여 입체감뿐 만아니라 상황에 맞는 사운드를 원하는 시간만큼 재생 할 수 있도록 하는 기술을 개발할 예정이다. 셋째, 음상정위가 포함된 음원 생성 기술개발로 음원 간 충돌 및 이벤트에 의해 발생하는 새로운 또는 비슷한 음원을 필터링을 적용하여 합성 생성하며 다양한 음향 효과를 연출할 수 있도록 하고 필요한 음원의 수를 최적화하는 기술의 개발을 추진할 예정이다.

궁극적으로 본 기술개발의 목표는 그래픽 콘텐츠개발자들이 사용하기 편리한 입체음 재생 인터페이스개발이다. 그러므로 본 연구의 논문에서 구현된 내용 및 향후 구현될 내용을 포함한 입체음 재생 인터페이스 기술개발을 지속적으로 추진할 것이다.

참 고 문 헌

- [1] 이영한, 김홍국, "가상현실 환경에서의 3차원 사운드 생성을 위한 거리 변화에 따른 구조적 머리전달함수 모델", 대한음성학회, pp.889-892, 2006. 01.
- [2] 김풍민 외5, "게임 소프트웨어를 위한 3차원 입체음향", 한국정보과학회 '97추계학술발표회, pp.332-334, 1997.
- [3] 김시호 외4, "HRTF를 이용한 헤드폰 기반의 다채널 입체음향 생성", 대한전자공학회, pp.72-73, 2005. 01.
- [4] 김관근, 김현태, 박장식, "입체음향 구현을 위한 머리전달함수의 IIR필터 설계", 한국콘텐츠학회, pp.341-344, 2005. 05.
- [5] 김현태, 박장식, 이광의, "스테레오 입체음향을 위한

- 머리 움직임 추정”, 멀티미디어학회, pp.1422-1424, 2005. 01.
- [6] 서울대학교 뉴미디어통신공동연구소, “2채널 스피커에 의한 3차원 음상정위 알고리즘 개발에 관한 연구”, 연구보고서, 1997.
- [7] Gardner, “W.G., 3G Audio Using Loudspeakers”, Kluwer Academic, Norwell, MA., pp.101-107, 1998.
- [8] D. R. Begault, “3-D Sound for Virtual Reality and Multimedia”, Academic Press, 1994.



원 용 태

2005년 : 전북대학교 영상공학과 (박사수료)
 2005년~현재 : 목포대학교 다도해 문화콘텐츠사업단 도서·해양문화콘텐츠연계전공 초빙교수

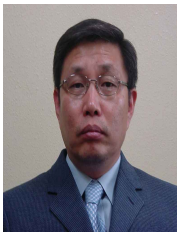
관심분야 : 영상처리, 가상현실(VR), Web3D 애니메이션



곽 훈 성

1979년 : 전북대학교 전자공학과 (박사)
 1981년~1982년 : 미국 텍사스 주립대학 연구교수
 1998년 : 과학기술법령정비정책위원

1994년~1995년 : 국가교육연구 전산망 추진위원
 1999년~현재 : 조달청우수제품(정보통신)심사위원
 1997년~현재 : (사)영상산업연구센터 대표
 현재 : 전북대학교 전자정보공학부 컴퓨터공학 교수 및 영상공학과(대학원) 주임교수
 관심분야 : 영상신호처리, 인공지능, 컴퓨터비전, 멀티미디어



장 봉 석

1997년 : 미국 메사추세츠주립대 전산학과 (박사)
 1997년~1999년 : 삼성전자 통신연구소 선임연구원

1999년~현재 : 목포대학교 정보공학부 교수
 관심분야 : 최적화, 센서망, 이동통신, 디지털콘텐츠



안 동 순

1985년 : 연세대학교 전자공학과 (박사)
 2002년~2003년 : 호주대학교 연구교수

1990년~현재 : 목포대학교 정보공학부 교수
 관심분야 : DSP, 임베디드시스템