

가상환경의 인터랙티브 사운드를 위한 'EVE-Sound™' 툴킷

남 양 희[†] · 성 숙 정^{**}

요 약

본 논문은 2D/3D 가상환경에서 실감 사운드의 실시간 합성을 위한 EVE-Sound™ 툴킷의 설계와 개발결과를 제시한다. EVE-Sound™는 사운드 효과 계산에 필요한 장면요소를 간추리는 전처리 도구와 실시간 렌더링 API들로 구성되며, 다중 모달리티의 실감 재현을 필요로 하는 가상현실, 게임 등의 각종 인터랙티브 콘텐츠에서 사용자에게 고급 그래픽 환경을 허용하면서 동시에 사운드의 실시간 재현을 가능하게 함을 목표로 하였다. 3차원 사운드 계산이 매우 복잡하기 때문에 기존의 사운드 API들이 간단한 사운드 효과와 재생을 제공하는 것과 달리, 복잡한 가상환경에서 3차원 사운드의 원리를 반영하되 그 렌더링을 실시간화하는 데 초점을 두었고, 이를 위해 새로운 장면 간략화 및 공간사운드 계산 방법이 제시되었다. 응용 사례 및 실험, 알고리즘 분석을 통해 제시된 툴킷의 활용성 및 향상된 실시간성을 확인하였다.

키워드 : 3차원사운드, 오디오-비주얼 렌더링, 가상현실, 사운드 라이브러리, 인터랙티브 콘텐츠

'EVE-Sound™' Toolkit for Interactive Sound in Virtual Environment

Yanghee Nam[†] · Sukjeong Sung^{**}

ABSTRACT

This paper presents a new 3D sound toolkit called EVE-Sound™ that consists of pre-processing tool for environment simplification preserving sound effect and 3D sound API for real-time rendering. It is designed so that it can allow users to interact with complex 3D virtual environments by audio-visual modalities. EVE-Sound™ toolkit would serve two different types of users: high-level programmers who need an easy-to-use sound API for developing realistic 3D audio-visually rendered applications, and the researchers in 3D sound field who need to experiment with or develop new algorithms while not wanting to re-write all the required code from scratch. An interactive virtual environment application is created with the sound engine constructed using EVE-Sound™ toolkit, and it shows the real-time audio-visual rendering performance and the applicability of proposed EVE-Sound™ for building interactive applications with complex 3D environments.

Key Words : 3D Sound, Audio-Visual Rendering, Virtual Environment, Sound Library, Interactive Contents

1. 서 론

실시간의 오디오-비주얼 렌더링은 3차원의 복잡한 가상환경의 실감 있는 재현을 위한 기본 요소이다. 미리 정해진 시나리오가 없는 인터랙티브 시스템에서는 시청각적 '사실감'도 중요하지만 실시간의 '상호작용성(응답성)'이 매우 중요하기 때문에 사용자에게 고급의 시청각 정보를 제시하면서도 인터랙션이나 상황 변화에 즉각적으로 피드백을 제공하는 것이 중요하다.

특히, 청각적 정보의 경우는 시각 정보보다 상황 전달력

이 빠른 경우가 많고, 장면이 어둡거나 시야에 없는 대상물의 상황을 전달하는 등 시각 정보의 부족 또는 부재시 이를 보완하고 부가 정보를 제공하는 역할도 하기 때문에 제2의 감각으로서 그 중요성이 적지 않다[1,2].

그럼에도 불구하고, 그래픽 정보에 대해서는 계산량이나 모델 복잡도를 줄이면서 사실감을 유지할 수 있는 방법들이 연구된 반면, 사운드에 대해서는 그러한 연구가 매우 적다.

이와 같이 3차원 사운드에 관한 연구나 활용이 미흡한 것은, 일반적으로 사용자가 그래픽 화면의 질의 변화에 보다 민감하기도 하지만, 3차원의 사운드를 복잡한 그래픽 환경에 추가 합성하여 재생하는 데 드는 계산량과 개발 노력이 매우 크기 때문이다. 특히, 빛과 사물과 사용자의 여러 속성에 따라 렌더링의 원리가 비교적 잘 알려진 그래픽스와 달리, 음원 및 청취자와 공간의 관계에 따른 음향의 전파, 흡

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음.

† 정 회 원 : 이화여자대학교 디지털미디어학부 부교수

** 준 회 원 : 이화여자대학교 디지털미디어학부 미디어공학전공 박사과정
논문접수: 2007년 5월 17일, 심사완료: 2007년 6월 11일

수 등의 원리는 음향학에서 흠어진 이론으로 존재하는 것이 대부분이어서 컴퓨터 개발자가 이를 찾아 분석하고 익혀서 활용하기까지 많은 어려움이 따른다.

따라서, 현재 소수의 사운드 프로그래밍 API가 제공되고는 있으나, 이들은 3차원의 공간특성이나 소리를 발생시키는 상황특성에 맞추어 가상 사운드를 렌더링 한다기 보다는 예견된 시나리오나 고정된 가상환경에 따라 미리 제작된 소리를 약간 조절하여 특정 타이밍에 트리거링(triggering)하게 하는 방식으로 사용되어 왔다.

그러나, 보다 실제감 있는 3차원 사운드 효과의 재현을 위해서는 실시간의 가상공간 특성과 사용자 이동에 따라 각 음원으로부터 청취자에게 전달되는 소리의 양을 계산하여 합성해야 한다. 이는 청취자의 이동시 공간감 생성에 필요한 전파, 반사, 간향 등을 매번 새롭게 계산함으로써 가능해지는데, 이 때 공간을 구성하는 모든 폴리곤들이 사운드의 전파나 반사 등에 영향을 끼치는 요인이 되기 때문에 복잡한 가상환경에 대한 3차원 사운드의 실시간 재현은 매우 많은 계산량을 요구하게 된다.

이에 본 논문에서는 기존의 API에서 가상환경의 구성이나 복잡도와 무관하게 사운드의 재생 기능에만 집중하였던 단점을 보완하고 임의의 3차원 가상환경에의 적용이 용이하도록 개선한 새로운 3D 사운드 구현 도구로서의 EVE-Sound™ 툴킷을 제시한다.

2. 범용 3차원 사운드 프로그래밍 도구

3차원 사운드 프로그래밍 분야에서 가장 널리 쓰이는 API는 DirectSound3D와 OpenAL이다[3][4]. DirectSound는 주로 게임과 같은 콘텐츠에서 현장감을 증대시키기 위해 사용되고 있으며, 다중 버퍼를 통해 하드웨어 출력을 지원하는 데 중점을 두고 있다. OpenAL은 DirectSound3D보다 다양한 플랫폼에서 비교적 상위레벨의 개념적 사운드 효과 프로그래밍을 가능하게 한다.

이 두 가지 도구 모두 3차원 공간 특성에 따른 사운드 효과를 위해서는 동굴, 운동장, 숲 속 등 몇 가지 종류의 환경에 대하여 미리 정해진 프리셋(preset) 사운드를 제공하고 있는데, 실제로는 같은 동굴이라도 크기, 모양, 장벽 등의 위치가 다르기 때문에 사실상 지나치게 단순화되어 있다고 할 수 있다. 그 외의 효과를 위해서는 사운드의 주파수 등의 성분을 조절할 수 있는 파라미터들을 제시하고 이를 조절하게 하고 있는데, 음향전문가가 아닌 개발자가 어떻게 파라미터 값들을 설정해야 원하는 사운드에 가까워지는지 알기 어렵다. 예컨대, 장애물, 장벽 등이 있는 경우 음의 전파 효과를 제어하기 위해 개발자로 하여금 고주파 또는 저주파 필터의 파라미터를 조절하게 하고 있는데, 이는 장벽의 두께나 흡음력과는 무관하게 개발자가 임의로 원하는 최종 결과를 상상하여 조절하게 하는 것이다. 따라서, 실제음과 유사한 가상음을 구축하기 위해서 많은 테스트 시간이 요구되며 개발자의 감각과 귀에 의존하게 된다.

이와 같이, 3차원 사운드 효과와 재생을 위한 범용 라이브러리들이 존재하기는 하지만 상당히 낮은 레벨의 기본적인 오디오 프로그래밍 인터페이스 기능을 제공하거나 반대로 아주 높은 수준의 제약적 상황설정과 시행착오(trial-and-error)에 의한 효과 시뮬레이션을 제공하는 데 그치고 있어서, 사운드 렌더러 개발자가 다양한 임의의 가상환경을 대상으로 공간 특성을 살린 사운드 효과를 만들기에 어려움이 많다.

3. EVE-Sound™ 툴킷

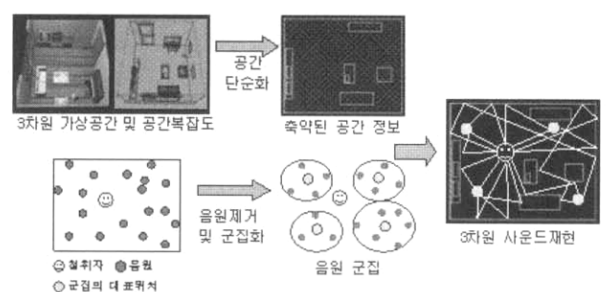
본 논문을 통해 제시하는 EVE-Sound™ 툴킷은 오디오와 그래픽의 실시간 동시 재현을 지원하며 네비게이션과 같은 인터랙션이 가능한 복잡한 가상환경에서 3차원 사운드 효과를 부여한 건축, 설계, 엔터테인먼트, 시뮬레이션 등의 응용 개발자를 위해 만들어진 사운드 렌더링 툴킷이다.

앞 장에서 설명된 바와 같이 기존의 3차원 사운드 API들은 3차원의 가상환경의 특성과는 무관하게 주파수 변형, 감쇠 제어 등 최종 음의 결과적 제어를 주 목적으로 하였다. 따라서, 이들 기존 API들만을 이용할 때는 개발자가 주어진 3차원의 가상환경을 이루는 구조와 구성물들에 관한 음향 효과 알고리즘을 직접 구현해야 하며, 이에 따라 사운드 파라미터를 적절히 조절하는 것도 개발자의 능력과 시간비용에 의존하게 된다.

반면, EVE-Sound™ 툴킷은 하부 출력 효과 제어를 담당하는 기존 API들과 함께 사용될 수 있으면서, 개발자로 하여금 상위 레벨에서 가상세계 음의 전파와 반사 등에 영향을 끼치는 다각형의 존재와 흡음 성질 등을 고려할 수 있게 하였기 때문에, 하부의 결과음을 미리 짐작하거나 일일이 설정하지 않더라도 가상환경의 성질에 대해 잘 정의하면 해당 환경에 맞는 사운드 효과가 자동 계산되게 한다.

3.1 EVE-Sound™ 툴킷의 주요 알고리즘

실시간 응용에서 3차원 가상환경의 음향적 특성을 렌더링하기 위한 EVE-Sound™의 주요 알고리즘은 공간 상에서의 음의 전파, 반사 등의 계산량을 실시간화하기 위한 공간 정보의 간략화 기법과, 간략화된 공간에서의 사운드 효과 계산 및 합성 방법으로 구성된다.



(그림 1) EVE-Sound™ 툴킷의 개요도

3.1.1 음향 공간 표현의 축약

사람이 듣는 소리는 음원(sound source)이 파장을 발생시켜 전달매개체(공기, 물 등)를 통해 사람에게 도달된 것으로서, 전파 과정에서 방해받지 않고 사람에게 직접 전달되는 것과 주위 물체에 반사되어 사람에게 도달하는 파장들이 합쳐진 것이다. 이 때 사람과 음원이 존재하는 환경의 특성, 즉, 공간의 크기나 복잡도, 공간 구성 물질, 전달 매개체 등의 속성에 따라 반사정도와 음원의 감쇠가 달라지므로 동일한 음원으로 구성되었다 할지라도 환경의 차이에 따라 다른 소리로 인지된다.

이와 같은 사운드 효과의 차이를 재현하기 위해서는 공간의 정보를 바탕으로 한 계산들이 필요하게 되는데, 일반적으로 그래픽 렌더링에 중심을 둔 가상환경 표현은 객체들의 기하 정보와 색상, 두께, 높이 및 텍스처, 위치, 방향 등의 정보로 구성된다. 그러나, 3차원 사운드의 계산에 이러한 모든 정보들이 이용되지는 않을 뿐더러, 그래픽 렌더링의 사실감을 위해 사용되는 모든 다각형들을 대상으로 사운드 전파, 반사 등의 계산을 한다면 실시간 계산량이 너무 막대해진다. 따라서, EVE-Sound™에서는 그래픽을 위한 가상공간 표현 외에 사운드 계산만을 위한 간략화된 공간 표현 구조를 생성하여 사용하도록 하였다.

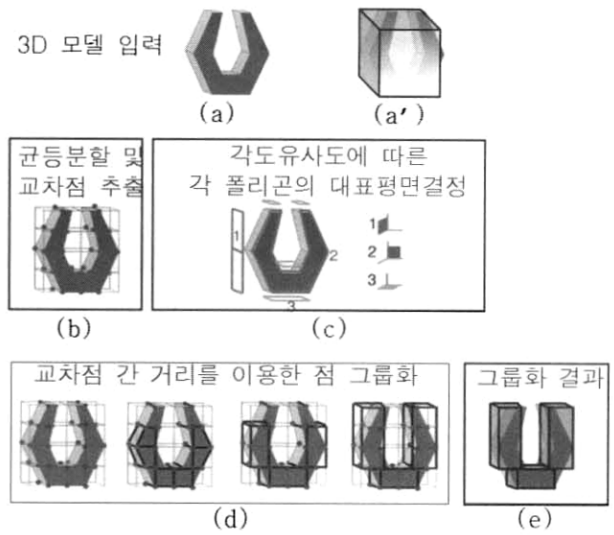
즉, EVE-Sound™에서 사운드 렌더링의 가속화에 적합한 공간 표현 구조를 생성하기 위해 ① 사운드에 영향을 주는 기하정보의 추출을 통한 공간 간략화 ② 실시간 렌더링을 돕기 위한 축약된 사운드 공간의 장면그래프 표현 생성 방법을 채택하였다.

그래픽에서 일반적으로 사용되는 기하정보의 단순화 기법과 달리, 사운드를 고려한 기하정보의 간략화 방법은 비교적 작은 구멍 등을 통해서도 전파되는 사운드의 특성 때문에 객체의 큰 틈새나 간격을 보존하는 방법을 구현하였다[5].

예컨대, '그래픽 렌더링 가속화'를 위한 기하 단순화에서는 (그림 2)의 (a)와 같이 가운데가 작게 패인 객체의 간략화 결과가 (그림 2)(a')와 같이 박스형으로 단순화되거나 타원체로 구해질 것이다. 그러나, '사운드 렌더링을 위한 기하 단순화'에서는 이와 같이 깊게 패인 홈이나 뚫린 공간이 있는 객체가 (그림 2)의 최종 결과 (e)와 같이 가장자리 면을 따라 형성된 몇 개의 바운딩 면의 집합으로 표현되어 소리가 전파될 수 있는 틈새를 보존하게 된다. 즉, 표현을 간략화하되 사운드 전파에 의미있는 구멍이나 틈새를 보존하는 것이다.

(그림 2)에서 보인 공간 축약 과정은 다음과 같다. 먼저, 단순화된 후의 면들이 몇 종류의 대표평면(예컨대, x,y,z평면)으로 표현될 지를 개발자로 하여금 지정하게 하고, (b)와 같이 대표평면과 나란한 일정 간격의 면들로 이루어지는 대표평면 그리드(grid)를 좁은 간격으로 생성한다.

그런 후, (c)와 같이 장면 내의 모든 다각형 면(facet)들을 지정된 대표평면들과의 각도 유사도를 통해 분류한다. 동시에, (b)에서 생성한 그리드와 각 객체의 면과의 교차점들을 구하여 그 점들을 오름차순으로 정렬하고 (d)와 같이



(그림 2) 축약 공간의 바운딩 면 생성 알고리즘

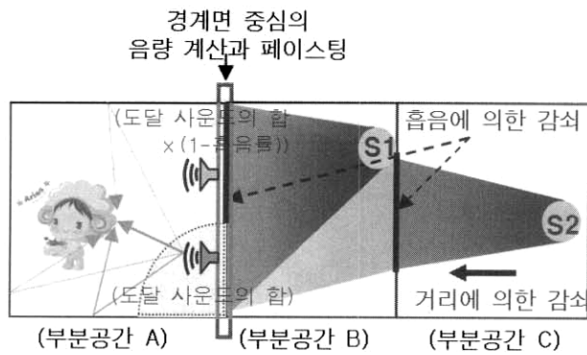
서로간의 거리를 검사하여 일정거리 이하일 경우 같은 그룹으로 표시한다. 이 때, 가까운 거리에 있는 교차점들의 집합이 서로 각 대표평면으로 감싸진 입방체 구조(cubic structure)를 유지하도록 하고, 이 조건을 만족하지 않으면 새로운 그룹으로 생성한다. 최종적으로, (e)와 같이 그룹별 바운딩 박스를 생성하고 생성된 바운딩 박스의 두께를 계산하여 일정범위 이상의 유효면적을 가지는 바운딩 면들로 축약 공간을 구성한다.

이와 같이 축약된 장면 정보는 그래픽 렌더링을 목적으로 한 형상 표현의 단순화 형태와는 달라지므로, EVE-Sound™에서는 생성된 바운딩 그룹들로 이루어지는 계층적 사운드 장면그래프 표현을 별도로 사용함으로써, 사운드 계산시 그래픽 장면그래프가 아닌 이 그래프를 참조하게 한다. 이 계층적 표현은 렌더링 할 대상 공간을 표현한다는 점에서는 기존의 장면 그래프(scenegrph)와 유사하지만 노드의 내용이 사운드 렌더링에 필요한 최소 정보를 표현하는 것으로서 논문[6]에 자세히 소개되었다.

사용자는 사운드 계산을 위한 장면 그래프를 원하는 구획에 따른 -대개 벽 등으로 둘러싸인 방이나 홀 등을 구획으로 하는-'부분 공간(sub-audiospace)'들로 구성할 수 있고, 각 부분 공간에 대해 공기나 물과 같은 전달매개체의 흡음률, 공간 크기, 해당 공간 내 음원들의 인덱스 및 중심위치 등의 속성을 설정하여, 장면 레벨에서 공간에 분포된 사운드를 설계하고 설정할 수 있다. 이 때 각 부분 공간을 구성하는 객체들의 기하정보는 앞서 설명한 기하정보 축약 방법에 따라 구해진 간략화된 바운딩 면들의 그룹들로 표현된다.

3.1.2 사운드 페이스팅과 실시간 합성

앞 절에서와 같이 사운드 효과와 연관된 가상환경의 표현 구조가 완성되면, 이를 이용하여 사운드의 전파, 반사, 잔향 등을 계산할 수 있다. 그러나, 바운딩 면들의 그룹들로 단순화되었다고는 해도 가상환경에 여러 음원들이 존재할 경우



(그림 3) 부분공간 경계면 중심의 사운드 계산

각 바운딩 면들에 대한 반사효과 및 청취자의 이동에 따른 각 음원의 전파 효과 등 3차원 사운드 효과를 실시간에 계산하는 것은 매우 어렵다.

이러한 문제를 다룬 연구는 최근 siggraph에 등장하기 시작했는데, Nicolas의 논문이 대표적이다[7]. 그러나, Nicolas의 방법은 음원들의 상호 근접도에 따라 무조건 지정된 갯수로 그룹화하기 때문에 청취자가 음원 가까이에서 귀를 대더라도 소리는 그룹의 대표음 위치에서 발생하는 것으로 들리게 되어 인지적 차이(gap)가 발생하게 된다. 또한, 특정 응용 유형에서는 실시간에 작동하지만 매 프레임마다 음원 그룹화를 수행하므로 알고리즘의 효율성이 떨어진다.

따라서, EVE-Sound™에서는 ‘사운드 페이스팅’이라는 새로운 음원 그룹화 방법을 제시하고 이를 통해 실시간 알고리즘 효율성을 개선하고자 하였다.

즉, 각 부분공간들을 감싸는 각 경계면에 대해 각 음원들로부터 해당 면에 직간접으로 도달하는 음의 양을 오프라인에서 미리 구하고, (그림 3)과 같이 각 경계면에 도달된 사운드들을 합하여 음을 ‘접착(페이스팅)’해놓음으로써, 실시간 계산량을 최소화하는 방법을 채택하였다. 이 때, 벽면과 같은 경계면에는 실제로 간략화된 바운딩 면(facet)이 존재하는 곳도 있고 창문과 같이 개방된 부분이 있게 되는데, 개방된 부분도 가상의 면으로서 단지 흡음률이 0이라는 차이가 있을 뿐 일반 면들과 동일하게 취급된다.

(그림 3)의 경우, 사용자가 부분공간 A에 있기 때문에 그 경계면을 중심으로 외부 음원 S1, S2로부터 도달하는 음량을 계산하게 된다. 이 때, 도달하는 음은 직접음일 경우 음원과 청취자 사이의 장벽과 전달매체의 흡음률에 따른 전파 효과와, 거리에 따른 감쇠 효과에 의해 결정되며, 반사음일 경우 반사되는 경계면의 흡음 속성에 따라 음의 크기가 변화된다. 그러나, 반사 계적을 실시간에 계산하는 것은 시간소모량이 크므로 이를 미리 계산하여 반사되어지는 가상 위치에 음원이 존재하는 것처럼 가상음원을 생성해놓는 이미시스 방법 사용하였다[8]. 이와 같이 하면, 경계면 중심의 도달 음량 계산시 반사를 따로 계산하지 않고 모든 가상과 실제 음원으로부터 직접 도달하는 직접음의 감쇠만 계산하면 된다.

이와 같이 각 부분공간의 경계면에 대해 도달되는 사운드

들을 미리 구해 페이스팅 해놓게 되면, 실시간에는 청취자가 어느 부분공간으로 이동하였느냐에 따라 해당 부분공간의 경계면과 청취자 사이의 지역적인 음의 전파 및 반사만을 계산하면 된다.

제안된 사운드 페이스팅 방법은 다수의 음원들이 부분공간 경계를 넘나들며 매우 빠르게 이동하는 경우에는 전파, 감쇠, 반사 등의 재계산량이 많으므로 알고리즘의 적용 효율이 크지 않지만, 음원 위치의 실시간 변화가 크지 않으면서 사용자가 주로 가상공간을 이동하게 되는 건축, 탐색 응용 등의 경우에는 실시간 렌더링 효율을 크게 높일 수 있다.

3.2 EVE-Sound™ 툴킷 및 API 구조

앞 절에서 기술된 알고리즘들을 바탕으로 한 EVE-Sound™ 툴킷은 (그림 4)와 같은 구조를 지닌다.

응용 프로그램 개발자는 오프라인의 음향공간 관리자리를 이용하여 가상환경 모델을 로딩하고, 앞 절에서 설명된 공간 정보 축약 알고리즘을 수행하는 관리도구 메뉴를 통해 간략화된 장면 표현을 생성한다. 축약된 공간정보에 대한 부분공간 경계 설정은, 이진 공간 분할에 의한 자동 설정법을 적용하거나, 개발자가 직접 경계면을 설정할 수 있다.

생성된 각 부분공간(audio-subspace)에 대해 환경 속성, 즉, 사운드의 전파나 잔향 등에 영향을 주는 공간의 크기, 전달 매개체(공기, 물 등) 등을 설정할 수 있으며, 유사한 사운드 환경 속성을 패턴화하여 저장했다가 다른 공간에 동일한 패턴을 부여할 수 있다. 끝으로, 가장 중요한 가상공간 음원의 위치와 음원 파일을 설정할 수 있다.

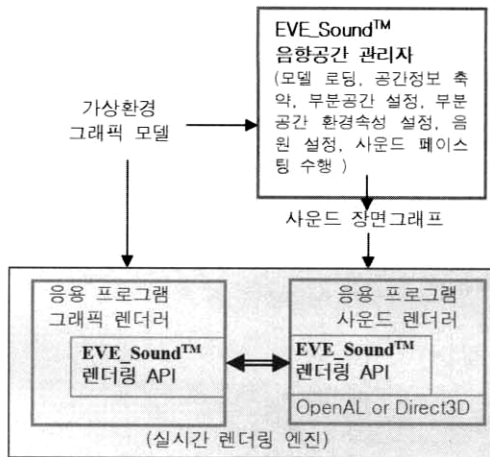
EVE_Sound™ 툴킷의 음향공간 관리자리를 이용한 오프라인 프로세스가 끝나면, 그 결과가 기록된 사운드 장면그래프가 생성되므로, 응용 프로그램 개발자는 실시간 렌더링 엔진에서 원하는 사운드 장면그래프를 로딩하면 된다.

(그림 5)는 EVE_Sound™의 오프라인 및 실시간 동작에 있어서 API 간 연관구조이며, (그림 6)은 관련된 주요 클래스의 구성을 보인다.

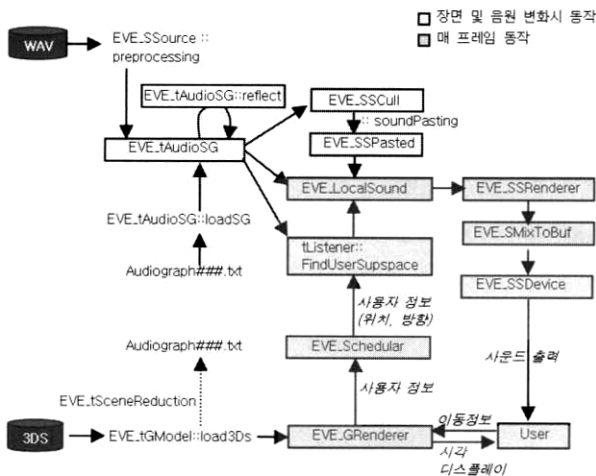
그래픽 모델에 대한 load3Ds함수는 3Ds 파일형식을 읽어 들여 3차원 장면을 구성하는 한편, EVE_tSceneReduction클래스를 통해 사운드 계산에 필요한 면 정보들만을 추출한다.

그 결과 간략화된 장면 정보는 텍스트 파일에 기록되며, 이에 대해 음향공간 관리자 프로그램을 통해 음원 및 부분공간 속성 정보를 부가할 수 있다.

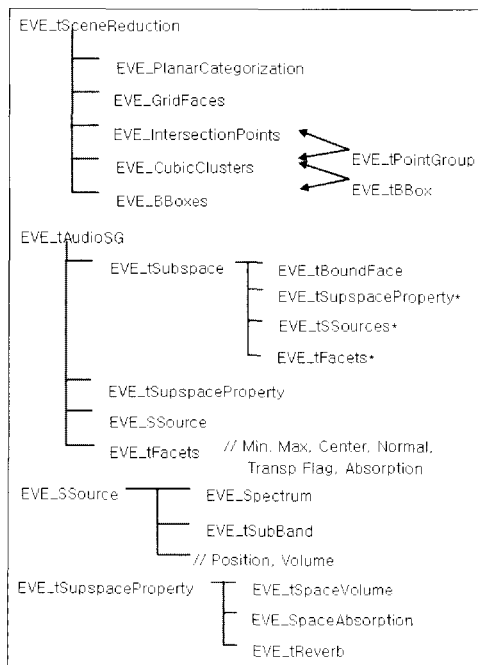
이 장면 정보 파일은 실시간 사운드 엔진에서 loadSG함수를 통해 접근 가능하며, 필요한 장면정보인 EVE_tAudioSG 객체를 구성한다. 이 객체에 포함된 음원들에 대해 간략화된 공간의 각 다각형에 대한 반사(reflection)를 계산하여 생성된 가상음원을 다시 EVE_tAudioSG객체의 음원 목록에 추가한다. EVE-Sound™ 툴킷은 부분공간 경계면(EVE_tBoundFace)으로부터 각 음원들까지의 거리 및 사용자 인지능력을 고려하여 인지되는 음원들만 남겨둘 수 있도록 음원 배제(culling) 작업을 수행한다(EVE_SSCull). 배제되지 않고 남은 음원들은 각 경계면에 도달하기까지의 전파



(그림 4) EVE_Sound™ 툴킷 활용 시스템 구조도



(그림 5) EVE_Sound™의 렌더링 API 동작구조



(그림 6) EVE_Sound™의 주요 클래스 구조

및 감쇠 등의 효과를 반영하여 각 경계면에 종속되어 기록되는 EVE_SSPasted 객체 목록을 형성한다.

위의 과정은 대개 로딩 시간에 수행되며, 실시간 인터랙션에서는 EVE_Scheduler를 통해 그래픽 및 사운드 렌더러가 사용자 정보의 변동과 프레임 동기화를 위한 최소 정보를 주고 받으며 일관성 있게 병렬 실행된다.

EVE_FindUserSubspace를 통해 장면 그래프 상에서 사용자(청취자)가 소속된 부분공간이 어디인지를 찾고, EVE_LocalSound클래스를 이용하여 해당 부분공간의 사방 경계면에 대해 페이스팅되어 있는 음원들(EVE_SSPasted)과 부분공간 내부 음원들의 영향만을 고려하여 결과음을 합성한다.

이 때, 양쪽 귀의 음량 차이, 부분공간 크기와 대개 특성에 의한 잔향효과, 도플러 효과 등의 처리를 거쳐 (EVE_SRenderer) 최종적으로 다중버퍼에 출력할 사운드들이 결정되어 출력된다.

(그림 5)에서 응용 프로그램 작성시 매 프레임 동작하는 부분은 회색 사각형으로 표시된 클래스들이다. 사용자가 비교적 변화가 없는 가상 공간을 이동(navigation)하는 경우, 제안된 API 동작구조에 따르면 사용자가 속한 부분공간의 변화 여부만 추적하여 해당 공간 내의 지역적 사운드 전파만을 계산하는 방식이 되기 때문에 빠른 렌더링이 가능하다.

사용자의 이동 외에 음원이 추가되거나 장면 내 변화가 생기는 경우, 즉 객체의 이동 및 삽입 등이 발생하는 경우에는 추가된 장면 객체들에 대해 사운드 장면 정보 상의 변화를 새로 계산하고 페이스팅 사운드를 새로 계산하여야 하지만, 변화가 한꺼번에 다량 발생하지 않는 일반적인 가상 환경에서는 성능의 급격한 저하가 일어나지 않는다.

사운드 렌더러는 C++ 기반의 간단한 class들로 구성되는 EVE-Sound™의 렌더링 API를 사용하여 3차원 사운드에 대한 이해가 없더라도 몇 줄의 간단한 코드로 3차원 렌더링을 지원할 수 있게 된다. 이 때, 개발자가 Direct3D나 OpenAL에 익숙하다면 필요한 파라미터의 직접 수정을 위해 이들 API들을 동시에 사용할 수도 있으나, 필요한 효과의 대부분이 EVE-Sound™에서 제공되기 때문에 그러한 필요성은 높지 않다.

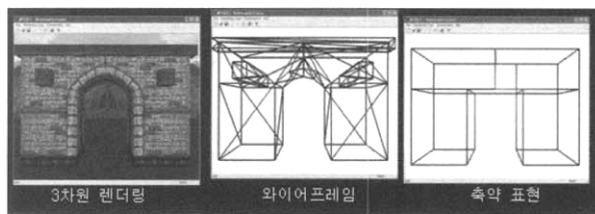
또한, 개발자가 각 단계의 알고리즘 중 일부를 변경하고자 한다면 해당 클래스를 파생하여 재작성할 수 있다.

4. 실험 및 평가

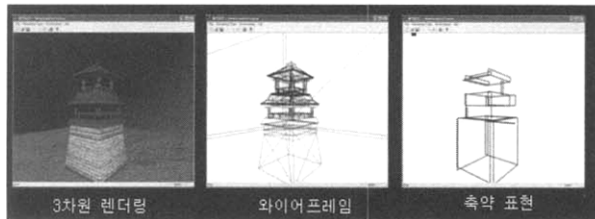
본 연구에서 제시한 EVE-Sound™의 API들은 가상 그래픽/사운드 렌더링의 실시간화를 목적으로 하였기 때문에 장면의 간략화 정도와 사운드 페이스팅에 의한 실시간 사운드 계산량의 축소를 통해 목적인 성능을 달성하고자 하였다. 이러한 성능의 확인을 위해 몇 가지 실험 및 응용 사례를 구축하여, 가상 장면정보 축약의 효과와 사운드 페이스팅을 통한 실시간 계산량 축소 효과 등을 확인하였다.

〈표 1〉 가상환경 장면 축약 실험

가상환경	면 개수		비고
	원본	축약후	
작은창이 있는 집	7751	66	큰 벽면에 의해 단순화되지 않고 작은창 보존
아치형문	736	30	작은상식 제거 문 보존((그림7)(a))
누각	660	32	기둥부분 뚫림 보존((그림7)(b))
전통가옥	306,477	274	



(a)



(b)

(그림 7) 장면 정보 축약 결과 예

4.1 장면 정보 간략화 실험

공개된 범용 그래픽 모델 두 종류와 본 연구팀의 응용 실험을 위해 구축된 복잡한 모델 등 4가지 가상환경에 대해 장면 정보 축약을 적용하였으며, <표 1>에 사례 실험 결과를 요약하였다.

(그림 7)의 (a)와 (b)는 실험 모델 가운데 두 사례이다. (a)에서는 아치형 문의 열린 통로가 보존되면서 736개 면이 30개로 축약되었는데, 아치형 문의 양쪽에 작은 육면체 구조물이 있으나 이는 사운드 통과가 가능한 구멍이나 틈새가 아니기 때문에 커다란 아치의 벽면과 일체화되어 단순화되었다. 반면, (b)에서는 2층으로 이루어진 누각의 기둥 사이 틈새가 크기 때문에 사운드가 통과하게 되고, 따라서 축약 후에도 이 부분의 뚫린 공간이 보존되었음을 알 수 있다.

4.2 실시간 사운드 계산성능 비교 분석

EVE_Sound™의 향상된 실시간 사운드 계산 성능을 보이기 위해, 현재 3차원 사운드의 실시간 계산능력 면에서 가장 우수하여 3.1.2절에서 언급된 Nicolas의 알고리즘[7]과 비교하였다(<표 2>).

```

for i=1 : 청취자가 속한 부분공간 내부 음원의 수 (n)
    내부음원 버퍼생성
    for j=1 : 부분공간의 경계면의 수 (b)
        가상음원생성; 가상음원버퍼생성(음원감쇠적용)
    end
end

// 1: 공간내부음원에 대한 가상음원수(l은 n보다 작거나 같다)
for i= 1 : n+l+b
    음원재생
end
    
```

(a) EVE_Sound™:O(n×b)

```

for i=1 : 전체음원의 수 (N)
    for j=1 : 전체 다각형수 (M)
        가상음원생성;
    end
end

for i=1 : 음원 군집의 수 (k)
    for j=1 : 전체음원의 수 (N)
        음원 군집 생성
    end
end

for k=1 :군집의 수(k)
    for j=1 : k번째 군집에 속한 음원의 수(s)
        버퍼생성
    end
end

for k=1 :군집의 수(k)
    음원재생
end
    
```

(b) Nicolas 방법[7] :O(N×M)

〈표 2〉 알고리즘 복잡도를 통한 실시간 계산량 비교

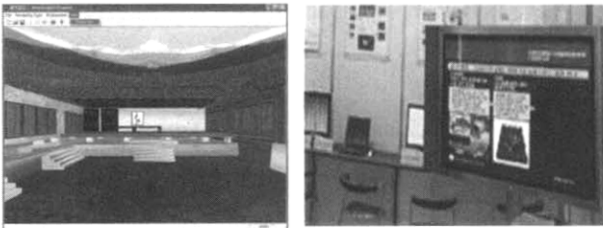
EVE_Sound™에서는 청취자가 속한 부분공간 내부의 음원들의 수(n) 만큼 각 경계면들(b)과의 반사를 계산하므로 <표 2>의 (a)와 같이 O(n × b)의 복잡도를 갖는다. 반면, Nicolas가 제시한 방법은 가상환경 전체의 음원들(N)과 다각형들(M)에 대해 반사를 계산하므로 알고리즘 복잡도가 O(N × M)이 된다. 이때, N 은 n 에 비해, M 은 b 에 비해 월등히 큰 수이며, 음원들의 군집도 실시간에 구해야 하기 때문에 음원의 수와 가상공간 복잡도가 증가할수록 계산량이 월등히 많아짐을 알 수 있다.

또한, EVE_Sound™에서는 음원의 수가 증가하더라도 사운드 페이스팅 등이 수행되는 전처리 시간이 증가할 뿐, 알고리즘의 특성상 특정 부분공간을 중심으로 한 내부 음원수는 거의 일정하므로, 실시간 계산량은 거의 증가하지 않는다.

4.3 응용 구축 및 인지 성능 실험

본 논문에서 제시한 EVE_Sound™ 툴킷의 기반 알고리즘들은 <표 1>에서 언급되었던 전통가옥 모델의 체험 응용 구축에 이용되었으며, ITRC Forum 2005에서 시연되었다. 이 시스템은 TCP/IP로 연결된 별도의 컴퓨터에서 그래픽 렌더링과 사운드 렌더링을 각각 수행하게 했으며, 그래픽 엔진은 1.7Ghz Intel Pentium 4, 1Gbyte DRAM과 FireGL 8800 환경에서, 사운드는 3.4Ghz Intel Pentium 4 CPU, 2Gbyte SRAM 환경에서 동작하였다.

이 실험 시스템은 풍악과 자연의 소리가 있는 한옥을 체험할 수 있게 하는 것으로서, 각 방에 기거하는 사람들의 특성에 따라 전통악기(거문고, 가야금 등) 연주소리가 들리고, 물소리, 벌레소리, 새소리 등 100여 개의 음원들이 동시에 소리를 낸다. 최종 출력 환경(스피커, 헤드폰 등)의 음질에 따라 이 응용의 체험자들이 네비게이션에 따른 소리변화의 방향을 확실히 인지하지 못하는 경우가 있었으나 가상 음원들의 위치나 거리를 전체적으로 확대하여 과장함으로써 음량의 차이를 증폭시키면 대부분의 사람들이 해당 소리가 접근과 후퇴, 좌/우 이동 등을 인지하여 해당 소리의 발원지를 올바르게 지적할 수 있었다.



(그림 8) 가상전통가옥 체험 응용 및 알고리즘 시연

5. 결론 및 향후 연구

본 논문에서는 소리와 영상이 동시에 제공되는 3차원의 인터랙티브 가상환경 응용 구축을 위해 공간감 있는 사운드의 실시간 재현방법에 초점을 둔 렌더링 알고리즘과 API를 포함한 툴킷을 제시하였다.

이에 대해, 본 논문에서 제시한 EVE-Sound™ 툴킷은 하위 레벨 제어가 중심이 되는 기존 API들과 달리 개발자의 제어 관점을 3차원 가상환경 공간에 대한 시청각 객체들의 배치와 구성을 중심으로 옮길 수 있도록 했다는 것이 가장 큰 차이라고 할 수 있으며, 그래픽과 동시 재현시 느린 속도의 극복을 위해 렌더링 효율화를 위한 장면 정보 간략화 및 사운드 페이스팅 기반의 성능개선 방법을 제시하였다.

EVE-Sound™ 는 다양한 복잡도의 가상환경 응용실험을 통해 가상환경의 복잡도에 크게 영향을 받지 않는 안정적인 실시간 사운드 렌더링 효과를 보였으며, 시각/청각 상호작용을 동시에 사용하게 함으로써 사용자에게 좀 더 가상환경을 자세히 인지할 수 있도록 도와줄 수 있을 것으로 기대한다.

참 고 문 헌

- [1] D. R. Begault, '3-D Sound for Virtual Reality and Multimedia,' Academic Press, London, 1994 (Revised in NASA/TM-2000-000000, Ames Research Center, 2000).
- [2] T. Funkhouser, "Computational Sound for Graphics, Virtual Reality, and Interactive Systems," SIGGRAPH Course Notes, 2002.
- [3] DirectSound 3D : Programming Guide in Microsoft Direct X, Microsoft, 2001.
- [4] 'OpenAL 1.1 Programmer's Guide,' Online Document, <http://www.openal.org>, 2005.
- [5] C. Joslin, Nadia M. Thalmann, "Significant Facet Retrieval for Real-Time 3D Sound Rendering in Complex Virtual Environments," Proceedings of the ACM Symposium on VR Software & Tech., pp.15-21, 2003.
- [6] J. S. Yi, S.J. Sung, Y. H. Nam, "The Realtime Method Based on Audio Scenegrph for 3D Sound Rendering", Lecture Notes in Computer Science, Vol.3767, Part I., pp.720-730, 2005.
- [7] Nicolas Tsingos, E. Gallo, G. Drettakis, "Perceptual Audio Rendering of Complex Virtual Environments," ACM Trans. on Graphics(SIGGRAPH Conference Proceedings), Vol. 23, No.3, pp.249-258, 2004.
- [8] J. Allen, D. Berkley, "Image Method for Efficiently Simulating Small Room Acoustics", Journal of Acoustical Society of America, Vol.65, No. 4, pp.943-951, 1979.



남 양 희

e-mail : yanghee@ewha.ac.kr

1989년 이화여자대학교 전자계산학과
(이학사)

1991년 KAIST 전산학과(공학석사)

1997년 KAIST 전산학과(공학박사)

1998년~1999년 스위스 로잔공대

박사후연구원

1999년~2000년 일본 ATR연구소 초청연구원

2000년~2002년 세종대학교 컴퓨터공학과 전임강사

2002년~2006년 이화여자대학교 디지털미디어학부 조교수

2006년~현재 이화여자대학교 디지털미디어학부 부교수

관심분야: 가상/증강현실, 영상처리, 모션인식, HCI



성 속 정

e-mail : monica810@ewhain.net

2002년 성결대학교 멀티미디어학과
(이학사)

2005년 이화여자대학교 대학원
디지털미디어학부(석사)

2007년~이화여자대학교 디지털미디어

학부 미디어공학전공 박사과정

관심분야: 가상/증강현실, 그래픽스, 영상처리, HCI