



(19) 대한민국특허청(KR)
 (12) 등록특허공보(B1)

(45) 공고일자 2018년01월31일
 (11) 등록번호 10-1824287
 (24) 등록일자 2018년01월25일

- (51) 국제특허분류(Int. Cl.)
G10L 19/00 (2006.01) *H04S 3/00* (2006.01)
 (21) 출원번호 10-2013-7011661
 (22) 출원일자(국제) 2011년10월26일
 심사청구일자 2016년10월25일
 (85) 번역문제출일자 2013년05월03일
 (65) 공개번호 10-2014-0000240
 (43) 공개일자 2014년01월02일
 (86) 국제출원번호 PCT/EP2011/068782
 (87) 국제공개번호 WO 2012/059385
 국제공개일자 2012년05월10일
 (30) 우선권주장
 10306211.3 2010년11월05일
 유럽특허청(EPO)(EP)
 (56) 선행기술조사문헌

Erik Hellerud, et al. Spatial redundancy in Higher Order Ambisonics and its use for lowdelay lossless compression. IEEE International Conference on Acoustics, Speech and Signal Processing. 2009. pp.26

(뒷면에 계속)

전체 청구항 수 : 총 11 항

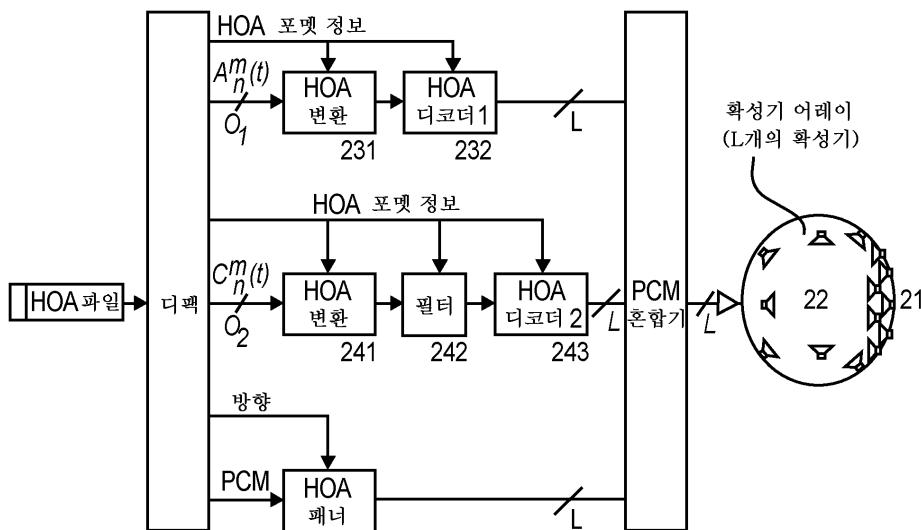
심사관 : 이남숙

- (54) 발명의 명칭 고차 앰비소닉 오디오 데이터를 위한 데이터 구조

(57) 요 약

본 발명은 고차 앰비소닉 HOA 오디오 데이터를 위한 데이터 구조에 관한 것으로, 이 데이터 구조는 하나 이상의 상이한 HOA 오디오 데이터 스트림 표현에 대한 2D 또는 3D 공간 오디오 콘텐츠 데이터를 포함한다. HOA 오디오 데이터는 '3'보다 큰 차수를 가질 수 있고, 이 데이터 구조는 또한, 고정되거나 시변동의 공간적 위치로부터의 단일 오디오 신호 소스 데이터 및/또는 마이크로폰 어레이 오디오 데이터를 포함할 수 있다.

대 표 도 - 도2



(72) 발명자

보엠, 요하네스

독일 30625 하노버 칼 비헤르트 알레 74 리서치 앤
드 이노베이션 도이치 톰슨 오에이치쥐

크롭, 홀거

독일 30625 하노버 칼 비헤르트 알레 74 리서치 앤
드 이노베이션 도이치 톰슨 오에이치쥐

바트케, 요한-마르쿠스

독일 30625 하노버 칼 비헤르트 알레 74 리서치 앤
드 이노베이션 도이치 톰슨 오에이치쥐

(56) 선행기술조사문헌

EP02205007 A1

KR1020060121807 A

JP2007500962 A

Daniel J, et al. Further investigations of
High Order Ambisonics and Wavefield Synthesis
for Holophonic Sound Imaging. 114th Auio
Engineering Society Convention. 2003.03.22.

명세서

청구범위

청구항 1

앰비소닉(Ambisonics) 계수들을 포함하는 고차 앰비소닉 HOA 오디오 데이터를 위한 데이터 구조를 포함하는 컴퓨터 판독가능 저장 매체로서,

상기 데이터 구조는 하나 이상의 상이한 HOA 오디오 데이터 스트림 표현들(audio data stream descriptions)을 위한 2D 및/또는 3D 공간 오디오 콘텐츠 데이터를 포함하고, 또한 상기 데이터 구조는 '3'보다 큰 차수를 갖는 HOA 오디오 데이터에 적합하며, 또한 데이터 구조는 고정되거나 시변동의 공간 위치들로부터의 단일 오디오 신호 소스 데이터 및/또는 마이크로폰 어레이 오디오 데이터를 포함할 수 있고,

상기 상이한 HOA 오디오 데이터 스트림 표현들은, 상이한 확성기 위치 밀도들, 코딩된 HOA 파동 타입들, HOA 차수들, 및 HOA 차원 중 적어도 2개와 관련되며,

하나의 HOA 오디오 데이터 스트림 표현은 프리젠테이션 장소(10)와 상이한 영역에 위치한 제1 확성기 배열(11, 21)을 갖는 프리젠테이션을 위한 오디오 데이터를 포함하고, 다른 HOA 오디오 데이터 스트림 표현은 상기 프리젠테이션 장소(10)를 둘러싸는 제2 확성기 배열(12, 22)을 갖는 프리젠테이션을 위한 오디오 데이터를 포함하며, 상기 제2 확성기 배열은 상기 제1 확성기 배열보다 낮은 확성기 위치 밀도를 갖는, 컴퓨터 판독가능 저장 매체.

청구항 2

제1항에 있어서,

상기 제1 확성기 배열(11, 21)에 대한 상기 오디오 데이터는 구면파(sphere wave)와 제1 앰비소닉 차수를 나타내고, 상기 제2 확성기 배열(12, 22)에 대한 상기 오디오 데이터는 평면파(plane wave) 및/또는 상기 제1 앰비소닉 차수보다 작은 제2 앰비소닉 차수를 나타내는, 컴퓨터 판독가능 저장 매체.

청구항 3

제1항에 있어서,

상기 데이터 구조는, 오디오 장면의 트랙들이 임의의 시간에서 시작 및 종료할 수 있는 장면 표현으로서 기능하는, 컴퓨터 판독가능 저장 매체.

청구항 4

제1항에 있어서,

상기 데이터 구조는,

- 청취 영역 외부 또는 내부의 오디오 소스들에 관련된 관심 영역;
- 구면 기초 함수(spherical basis function)들의 정규화;
- 전파 방향성;
- 앰비소닉 계수 스케일링 정보;
- 앰비소닉 파동 타입, 예를 들어, 평면 또는 구면;
- 구면파의 경우, 디코딩을 위한 기준 반경

에 관한 데이터 항목들을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 5

제1항에 있어서,

상기 앰비소닉 계수들은 복소 계수(complex coefficient)들인, 컴퓨터 판독가능 저장 매체.

청구항 6

제1항에 있어서,

상기 데이터 구조는 하나 이상의 마이크로폰에 대한 방향들과 특성들에 관한 메타데이터를 포함하고, 및/또는 단일-소스 입력 신호들에 대한 적어도 하나의 인코딩 벡터(encoding vector)를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 7

제1항에 있어서,

상기 앰비소닉 계수들의 적어도 일부는 대역폭이 감소되어, 상이한 HOA 차수들에 대해, 관련된 앰비소닉 계수들의 대역폭이 상이한(1221-122N), 컴퓨터 판독가능 저장 매체.

청구항 8

제7항에 있어서,

상기 대역폭의 감소는 MDCT 처리(1431-143M)에 기초하는, 컴퓨터 판독가능 저장 매체.

청구항 9

삭제

청구항 10

오디오 프리젠테이션을 위한 방법으로서,

적어도 2개의 상이한 HOA 오디오 데이터 신호를 포함하는 HOA 오디오 데이터 스트림을 수신하는 단계;

이들 중 적어도 제1 신호를 프리젠테이션 장소(10)와 상이한 영역에 위치한 제1 확성기 배열(11, 21)을 갖는 프리젠테이션을 위해 이용하는 단계(231, 232); 및

이들 중 적어도 제2의 다른 신호를 상기 프리젠테이션 장소(10)를 둘러싸는 제2 확성기 배열(12, 22)을 갖는 프리젠테이션을 위해 이용하는 단계(241, 242, 243)

를 포함하고,

상기 제2 확성기 배열은 상기 제1 확성기 배열보다 낮은 확성기 위치 밀도를 갖는, 방법.

청구항 11

제10항에 있어서,

상기 제1 확성기 배열(11, 21)에 대한 상기 오디오 데이터는 구면파들과 제1 앰비소닉 차수를 나타내고, 상기 제2 확성기 배열(12, 22)에 대한 상기 오디오 데이터는 평면파들 및/또는 상기 제1 앰비소닉 차수보다 작은 제2 앰비소닉 차수를 나타내는, 방법.

청구항 12

오디오 프리젠테이션을 위한 장치로서,

적어도 2개의 상이한 HOA 오디오 데이터 신호를 포함하는 HOA 오디오 데이터 스트림을 수신하기 위한 수단; 상기 신호들 중 적어도 제1 신호를 프리젠테이션 장소와 상이한 영역에 위치한 소정의 확성기 배열을 갖는 프리젠테이션을 위해 처리하기 위한 수단; 및 상기 신호들 중 제2의 다른 신호를 상기 프리젠테이션 장소를 둘러싸는 상이한 확성기 배열을 갖는 프리젠테이션을 위해 처리하기 위한 수단을 포함하고, 상기 상이한 확성기 배열은 상기 소정의 확성기 배열보다 낮은 확성기 위치 밀도를 갖는, 장치.

청구항 13

작제

발명의 설명

기술 분야

[0001] 본 발명은, 2D 및/또는 3D 공간 오디오 콘텐츠 데이터를 포함하며, '3'보다 큰 차수를 갖는 HOA 오디오 데이터에도 적합한 고차 Ambisonics 오디오 데이터에 대한 데이터 구조에 관한 것이다.

배경 기술

[0002] 3D 오디오는 이하에서 설명되는 고차 앰비소닉(HOA)이라 불리는 기술에 의한 음장 표현(sound field description)을 이용하여 실현될 수 있다. HOA 데이터를 저장하는 것은, 주어진 재생 스피커 셋업에서의 재생을 위해 확성기 신호를 생성할 수 있는 특별한 디코더에 의해 이 데이터가 어떻게 이용되어야 하는지에 관한 소정의 규약과 조항을 요구한다. 어떠한 기준의 저장 포맷도 HOA에 대한 이들 조항들 모두를 정의하지 않는다. 예를 들어, Martin Leese의 "File Format for B-Format", <http://www.ambisonia.com/Members/etienne/Members/mleese/file-format-for-b-format>에서 2009년 3월 30일자로 기술된 바와 같은 *.amb 파일 포맷 구현을 갖는 (확장가능한 'Riff/wav'구조에 기초한) B-Format은 오늘날 이용가능한 가장 정교한 포맷이다. 그 .amb 파일 포맷은 2000년도 ICMC 베를린 2000에서, R.W. Dobson의 "Developments in Audio File Formats"에서 소개되었다.

[0003] 2010년 7월 16일자로, 기존 파일 포맷의 개요가 Xchange Site: "Existing formats", <http://ambisonics.iem.at/xchange/format/existing-formats>에 공개되어 있고, 앰비소닉 교환 포맷을 위한 제안도 역시 이 사이트에 공개되어 있다: "A first proposal to specify, define and determine the parameters for an Ambisonics exchange format", <http://ambisonics.iem.at/xchange/format/a-first-proposal-for-the-format>.

[0004] HOA 신호에 관련하여, 모두 동일한 주파수의 상이한 음향 소스로부터의, 3D의 경우 $M = (N + 1)^2$ 개의 (2D의 경우 $(2N + 1)$ 개의) 상이한 오디오 객체들의 집합이 공간적으로 균등하게 분포되어 있다면, 레코딩되어(인코딩되어) 상이한 음향 객체로서 재생될 수 있다. 이것은, 1차 앰비소닉 신호는 4개의 3D 또는 3개의 2D 오디오 객체를 운반할 수 있고 이들 객체들은 3D의 경우에는 구(sphere) 주변에, 2D의 경우에는 원(circle) 주변에 균일하게 분리될 필요가 있다는 것을 의미한다. 레코딩시에 공간적 중첩과 M개 보다 많은 신호들의 결과 흐릿해짐(blur)이 발생할 것이다 - 가장 시끄러운 신호들만 코히어런트 객체(coherent object)로서 재생될 수 있고, 다른 확산 신호들은, 공간에서의 중첩, 주파수 및 라우드니스 유사도(loudness similarity)에 따라 코히어런트 신호들을 다소 퇴보(degenerate)시킬 것이다.

[0005] 영화관의 음향 상황에 관련하여, 정면 스크린 영역에 대해서는 시각적 장면과 매칭되기 위하여 높은 공간적 음향 위치파악 정확성(sound localisation accuracy)이 요구된다. 주변 음향 객체(반향음, 시각 장면과 연계되지 않은 음향 객체)의 인지는 덜 중요하다. 여기서 스피커들의 밀도는 정면 영역에 비해 더 작을 수 있다.

[0006] 정면 영역에 관련된 HOA 데이터의 HOA 차수는, 선택시 홀로포닉 재생(holophonic replay)을 가능케하도록 충분히 클 필요가 있다. 전형적인 차수는 N=10이다. 이것은 $(N + 1)^2 = 121$ 개의 HOA 계수를 요구한다. 이론적으로, 우리는 M=121개의 오디오 객체들을, 만일 이 오디오 객체들이 공간적으로 균등하게 분포될 것이라면, 인코딩할 수 있다. 그러나, 우리의 시나리오에서, 이들은 정면 영역으로 제약된다(여기서에만 우리가 이러한 높은 차수를 필요로 하기 때문이다). 사실상, 우리는 흐릿해짐이 없이 약 M=60개의 오디오 객체만을 코딩할 수 있다(정면 영역은 기껏해야 방향구(sphere of direction)의 절반, 따라서 M/2).

[0007] 전술된 B-Format에 관련하여, 이 포맷은 앰비소닉 차수 3까지만의 표현을 가능케하고, 파일 크기는 4GB로 제약된다. 현대의 디코더에 필수적인 파동 타입이나 기준 디코딩 반경과 같은, 다른 특별한 정보 항목들은 빠져 있다. 상이한 앰비소닉 컴포넌트들(채널들)에 대해 상이한 샘플 포맷들(워드 폭들) 및 대역폭을 사용하는 것은 가능하지 않다. 앰비소닉에 대한 부가 정보와 메타데이터를 저장하기 위한 표준도 역시 없다.

[0008] 공지된 기술에서, 마이크로폰 어레이를 이용하여 앰비소닉 신호를 레코딩하는 것은 차수 1로 제약된다.

이것은, HOA 마이크로폰의 실험적 프로토타입이 개발된다면 향후에 변경될 수도 있다. 3D 콘텐츠의 생성을 위해, 마이크로폰 어레이를 이용하여 주변 음장(ambience sound field)의 표현이 1차 앰비소닉으로 레코딩될 수 있고, 이로써 클로즈-업 모노 마이크로폰 또는 고지향성 마이크로폰을 이용하여 방향 소스가 방향 정보(즉, 소스의 위치)와 함께 포착된다. 그 다음 방향 신호가 HOA 표현 내에 인코딩되거나, 이것은 정교한(sophisticated) 디코더에 의해 수행될 수도 있다. 어쨌든, 새로운 앰비소닉 파일 포맷은 한 번에 하나보다 많은 음장 표현을 저장할 수 있어야 할 필요성이 있지만, 어떠한 기준의 포맷도 하나보다 많은 앰비소닉 표현을 포함시킬 수 있을 것 같지 않다.

발명의 내용

해결하려는 과제

[0009]

본 발명에 의해 해결될 문제는, 한 번에 2개 이상의 음장 표현을 저장할 수 있는 앰비소닉 파일 포맷을 제공하는 것이며, 여기서, 앰비소닉 차수는 3보다 클 수 있다. 이 문제는 청구항 1에 개시된 데이터 구조와 청구항 12에 개시된 방법에 의해 해결된다.

과제의 해결 수단

[0010]

현실적인 3D 오디오를 재현하기 위해, 차세대 앰비소닉 디코더들은, 처리될 저장된 데이터와 함께 많은 규약과 조항을 요구하거나, 모든 관련 파라미터들과 데이터 요소들이 일관적으로 저장될 수 있는 단일 파일 포맷을 요구할 것이다.

[0011]

공간 음향 콘텐츠를 위한 본 발명의 파일 포맷은 하나 이상의 HOA 신호 및/또는 방향 모노 신호(directional mono signal)를 방향 정보와 함께 저장할 수 있고, 3보다 큰 앰비소닉 차수와 4GB보다 큰 파일이 실현가능하다. 또한, 본 발명의 파일 포맷은 기준 포맷들이 제공하지 않는 추가 요소들을 제공한다:

[0012]

1) 차세대 HOA 디코더에 요구되는 필수 정보가 이 파일 포맷 내에 저장된다:

[0013]

- 앰비소닉 파동 정보(평면, 구면, 혼합형), 관심 영역(청취 영역 바깥 또는 이내의 소스), 및 (구면파의 디코딩을 위해) 기준 반경

[0014]

- 관련된 방향 모노 신호가 저장될 수 있다. 이를 방향 신호들의 위치 정보는, 각도 및 거리 정보를 이용하거나 앰비소닉 계수들의 인코딩 벡터를 이용하여 기술될 수 있다.

[0015]

2) 앰비소닉 데이터를 정의하는 모든 파라미터들이, 레코딩에 대한 명료성을 보장하기 위해 부가 정보 내에 포함된다:

[0016]

- 앰비소닉 스케일링 및 정규화(SN3D, N3D, Furse Malham, B Format, …, 사용자 정의형), 혼합된 차수 정보.

[0017]

3) 앰비소닉 데이터의 저장 포맷은 데이터의 융통성 있고 경제적인 저장을 허용하도록 확장된다:

[0018]

- 본 발명의 포맷은, 제약된 대역폭을 이용하는 것 뿐만 아니라 상이한 PCM-워드(word) 크기 해상도를 갖는 앰비소닉 차수에 관련된 데이터 저장(앰비소닉 채널)을 허용한다.

[0019]

4) 메타 필드는, 마이크로폰 신호에 대한 레코딩 정보와 같은 파일에 관한 동반 정보의 저장을 허용한다:

[0020]

- 레코딩 기준 좌표계, 마이크로폰, 소스 및 가상 청취자 위치, 마이크로폰 방향 특성, 룸 및 소스 정보.

[0021]

2D 및 3D 오디오 콘텐츠에 대한 이 파일 포맷은, 고정적이거나 시변동 위치를 갖는 단일 소스 뿐만 아니라 고차 앰비소닉 표현(HOA)의 저장을 커버하며, 차세대 오디오 디코더들이 현실적인 3D 오디오를 제공할 수 있게 하는 모든 정보를 포함한다.

[0022]

적절한 설정을 이용하면, 본 발명의 파일 포맷은 오디오 콘텐츠의 스트리밍에도 적합하다. 따라서, 콘텐츠-의 존적 부가 정보(헤더 데이터)는 파일의 생성자(creator)에 의해 선택된 시점에 전송될 수 있다. 본 발명의 파일 포맷은 또한, 오디오 장면의 트랙들이 임의의 시간에 시작하고 끝날 수 있는 장면 표현으로서도 역할한다.

[0023]

원칙적으로, 본 발명의 데이터 구조는 고차 앰비소닉 HOA 오디오 데이터에 적합하며, 이 데이터 구조는 하나 이상의 상이한 HOA 오디오 데이터 스트리밍 표현을 위한 2D 및/또는 3D 공간 오디오 콘텐츠 데이터를 포함하고, 이 데이터 구조는 또한, '3'보다 큰 차수를 갖는 HOA 오디오 데이터에 적합하며, 이 데이터 구조는 또한 고정되어 시변동 공간 위치로부터의 단일 오디오 신호 소스 데이터 및/또는 마이크로폰 어레이 오디오 데이터를 포함

할 수 있다.

[0024] 원칙적으로, 본 발명의 방법은 오디오 프리젠테이션에 적합하며, 여기서, 적어도 2개의 상이한 HOA 오디오 데이터 신호를 포함하는 HOA 오디오 데이터 스트림이 수신되고, 이들 중 적어도 제1 하나는 프리젠테이션 장소의 별개 영역에 위치한 밀집한 확성기 배열로 프리젠테이션하는데 이용되고, 이들 중 적어도 제2의 상이한 하나는 상기 프리젠테이션 장소를 둘러싸는 덜 밀집한 확성기 배열로 프리젠테이션하는데 이용된다.

[0025] 본 발명의 유익한 추가 실시예들이 각 종속항들에 개시된다.

도면의 간단한 설명

[0026] 본 발명의 예시적 실시형태들이 첨부된 도면을 참조하여 설명되며, 여기서,

도 1은 정면 영역에서는 밀집한 스피커 배열을 갖고 청취 영역을 둘러싸는 덜 밀집한 스피커 배열을 갖는 영화관에서의 홀로포닉 재생;

도 2는 정교한 디코딩 시스템;

도 3은 마이크로폰 어레이 레코딩, 단일 소스 레코딩, 간이 및 복합 음장 생성으로부터의 HOA 콘텐츠 생성;

도 4는 차세대 몰입형 콘텐츠(immersive content) 생성;

도 5는 간이 서라운드 확성기 셋업에 대한 HOA 신호의 2D 디코딩과, 정면 스테이지용 홀로포닉 확성기 셋업 및 덜 밀집한 3D 서라운드 확성기 셋업에 대한 HOA 신호의 3D 디코딩;

도 6은 내부 도메인 문제, 여기서, 소스는 관심/유효 영역 바깥에 있다;

도 7은 구면 좌표의 정의;

도 8은 외부 도메인 문제, 여기서, 소스는 관심/유효 영역 내부에 있다;

도 9는 간단한 예시적 HOA 파일 포맷;

도 10은 복수의 트랙을 갖는 복수 프레임을 포함하는 HOA 파일에 대한 예;

도 11은 복수의 MetaDataChunk들을 갖는 HOA 파일;

도 12는 TrackRegion 인코딩 처리;

도 13은 TrackRegion 디코딩 처리;

도 14는 MDCT 처리를 이용한 대역폭 감소의 구현;

도 15는 MDCT 처리를 이용한 대역폭 복원의 구현.

발명을 실시하기 위한 구체적인 내용

[0027] 3D 비디오의 계속적 확산과 더불어, 몰입형 오디오 기술이 차별화된 관심 대상이 되고 있다. HOA(Higher Order Ambisonics)는, 영화관에 3D 오디오를 점진적으로 도입하는 방식을 제공할 수 있는 이들 기술들 중 하나이다. HOA 사운드 트랙 및 HOA 디코더를 이용하여, 영화관은 기존의 오디오 서라운드 스피커 셋업으로 시작하여, 단계별로 더 많은 확성기에 투자하여, 각 단계에서 몰입형 경험을 개선시킨다.

[0028] 도 1a는 정면 영역에서는 밀집한 확성기 배열(11)을 갖고 청취 또는 좌석 영역(10)을 둘러싸는 덜 밀집한 확성기 밀도(12)를 가짐으로써, 시각적 액션에 관련된 음향의 정확한 재생 방식과, 재생된 주변 음향의 충분한 정확성을 제공하는 영화관의 홀로포닉 재생을 도시한다.

[0029] 도 1b는 재생된 정면 음파 도달의 인지된 방향을 도시하며, 여기서, 평면파의 도달 방향은 상이한 스크린 위치와 정합한다, 즉, 평면파는 깊이감을 재생하기에 적합하다.

[0030] 도 1c는, 스크린 주변에서 인지된 음향 방향과 3D 시각적 액션의 더 나은 일치성으로 이어지는, 재생된 구면파 도달의 인지된 방향을 도시한다.

[0031] 영화관에서의 주요 시각적 액션은 청취자의 정면 영역에서 발생한다는 사실에서 2개의 상이한 HOA 스트림에 대한 필요성이 생긴다. 또한, 주변 소스의 경우보다 정면 음향 소스의 경우에 음향의 방향 검출의 인지적 정확성이 더 높다. 따라서, 정면 음향 재생의 정확성이 재생된 주변 음향에 대한 공간적 정확성보다 더 높을 필요가

있다. 정면 스크린 영역에 대해서는, 음향 재생을 위한 홀로포닉 수단, 많은 수의 확성기, 전용 디코더 및 관련된 스피커 드라이버가 요구되는 반면, 주변 음향 재생에 대해서는 덜 비싼 기술이 필요하다(청취 영역을 둘러싼 스피커의 더 낮은 밀도와 디코딩 기술의 더 낮은 정확성).

[0032] 콘텐츠 생성 및 음향 재생 기술 때문에, 주변 음향에 대해 하나의 HOA 표현을 제공하고 전경 액션 음향에 대해 하나의 HOA 표현을 제공하는 것이 유익하다. 도 4를 참조한다. 간단한 대장의 재생 음향 장비를 갖춘 간단한 셋업을 이용한 영화관은 디코딩 이전에 양쪽 스트리밍을 혼합할 수 있다(도 5의 위쪽 참조). 완전 몰입형 재생 수단을 탑재한 더 정교한 영화관은 2개의 디코더를 이용할 수 있다 - 도 2의 정교한 디코딩 시스템과 도 5의 하단부에 도시된 바와 같이, 하나는 주변 음향 디코딩을 위한 것이고, 또 하나의 전문화된 디코더는 전경 주요 액션에 대한 가상 음향 소스의 고정밀 위치파악을 위한 것이다.

[0033] 특별 HOA 파일은, 주변 음향 $A_n^m(t)$ 및 시각적 주요 액션에 관련된 정면 음향 $C_n^m(t)$ 에 대한 HOA 음장을 나타내는 적어도 2개의 트랙을 포함한다. 방향 효과를 위한 선택사항적 스트리밍이 제공될 수도 있다. 팬너(panner)와 함께 2개의 대응하는 디코더 시스템은 밀집한 정면 3D 홀로포닉 확성기 시스템(21)과 덜 밀집한(즉, 드문드문 있는) 3D 서라운드 시스템(22)을 위한 신호를 제공한다.

[0034] Track 1 스트리밍의 HOA 데이터 신호는 주변 음향을 나타내고, 주변 음향 재생 전용의 디코더1(232)로의 입력을 위해 HOA 변환기(231)에서 변환된다. Track 2 데이터 스트리밍의 경우, HOA 신호 데이터(시각적 장면에 관련된 정면 음향)는, 전용 디코더2(243)와 더불어 스크린 영역 주변에 구면 음향 소스의 최상의 배치를 위한 거리 교정(수학식 26) 필터(242)로의 입력을 위해 HOA 변환기(241)에서 변환된다. 방향 데이터 스트리밍은 L 스피커에 직접 페닝(panning)된다. 3개의 스피커 신호들이 3D 스피커 시스템에 의한 합동 재생을 위해 PCM 혼합된다.

[0035] 이러한 시나리오 전용의 공지된 파일 포맷은 없는 것 같다. 공지된 3D 음장 레코딩은, 이후의 재생을 위한 저장시에, 관련된 음향 트랙을 갖는 전체 장면 표현을 이용하거나 단일 음장 표현을 이용한다. 제1 종류의 예는, WFS(Wave Field Synthesis) 포맷과 다양한 컨테이너 포맷이다. 제2 종류의 예는, B 또는 AMB 포맷과 같은 앰비소닉 포맷으로, 전술된 문헌 "File Format for B-Format"을 참조한다. 후자는, 앰비소닉 차수 3, 고정 전송 포맷, 고정 디코더 모델 및 단일 음장으로 제한된다.

[0036] HOA 콘텐츠 생성 및 재생

[0037] HOA 음장 표현을 생성하기 위한 처리가 도 3에 도시되어 있다.

[0038] 도 3a에서, 음장의 자연 레코딩(natural recording)이 마이크로폰 어레이를 이용하여 생성된다. HOA 신호를 형성하기 위해 캡슐 신호가 행렬화되고 등화된다. 고차 신호들(앰비소닉 차수 > 1)은 대개 대역-통과 필터링되어 캡슐 거리 효과로 인한 아티팩트를 줄인다: 저역-통과 필터링되어 고주파수에서의 공간 에일리어스를 줄이고, 고역-통과 필터링되어 앰비소닉 차수 n의 증가($h_n(kr_{d_mic})$, 수학식 34 참조)에 따른 과도한 저주파 레벨을 줄인다. 선택사항으로서, 거리 코딩 필터링이 적용될 수 있는데, 수학식 25 및 27을 참조한다. 저장 이전에, HOA 포맷 정보가 트랙 헤더에 추가된다.

[0039] 예술 음장 표현(artistic sound field representation)은 대개, 복수 방향 단일 소스 스트리밍을 이용하여 생성된다. 도 3b에 도시된 바와 같이, 단일 소스 신호는 PCM 레코딩으로서 포착될 수 있다. 이것은, 클로우즈업 마이크로폰에 의해 또는 고지향성의 마이크로폰을 이용하여 이루어질 수 있다. 가상의 최상 청취 위치(HOA 좌표계, 또는 이후의 맵핑을 위한 임의의 기준점)에 관한 음향 소스의 방향 파라미터(r_s, Θ_s, ϕ_s)가 레코딩된다. 영화 장면의 렌더링시에 음향을 예술적으로 배치함으로써 거리 정보도 역시 생성될 수 있다. 도 3c에 도시된 바와 같이, 인코딩 벡터(Ψ)를 생성하기 위해 방향 정보(Θ_s, ϕ_s)가 이용되고, 방향 소스 신호가 앰비소닉 신호 내로 인코딩되는데, 수학식 18을 참조한다. 이것은 평면과 표현과 등가이다. 말단의 필터링 프로세스는, 구면 소스 특성을 앰비소닉 신호 내에 새기거나(수학식 19), 거리 코딩 필터링을 적용하기 위해(수학식 25 및 27), 거리 정보 r_s 를 이용할 수 있다. 저장 이전에, HOA 포맷 정보가 트랙 헤더에 추가된다.

[0040] 도 3d에 도시된 바와 같이 앰비소닉 신호를 HOA 혼합함으로써 더 복잡한 파동장 표현이 생성된다. 저장 이전에, HOA 포맷 정보가 트랙 헤더에 추가된다.

[0041] 3D 영화관을 위한 콘텐츠 생성의 프로세스가 도 4에 도시되어 있다. 시각적 액션에 관련된 정면 음향은 높은

공간 정확도로 인코딩되어 HOA 신호(파동장)($C_n^m(t)$)와 혼합되고 Track 2로서 저장된다. 연관된 인코더들은, 시각적 장면과의 최상의 매칭에 필요한 특별한 파동 타입과 높은 공간적 정밀도로 인코딩한다. Track1은 음향

방향의 제약없이 인코딩된 주변 음향에 관련된 음장($A_n^m(t)$)을 포함한다. 대개는 주변 음향의 공간적 정확도는 정면 음향의 경우만큼 높을 필요는 없고 (결과적으로 앰비소닉 차수는 더 작을 수 있고), 파동 타입의 모델링이 덜 중요하다. 주변 음장은 또한, 정면 음향 신호의 반향 부분을 포함할 수 있다. 양쪽 트랙들은 저장 및/또는 교환을 위해 멀티플렉싱된다.

[0042] 선택사항으로서, 방향 음향(예를 들어, Track 3)이 파일에 멀티플렉싱될 수 있다. 이를 음향은, 특수 효과 음향, 시각적으로 손상을 받는 대화 또는 스포츠 해설과 같은 정보일 수 있다.

[0043] 도 5는 디코딩의 원리를 도시한다. 위쪽 부분에 도시된 바와 같이, 드문드문한 확성기 셋업을 갖는 영화관은, 간소화된 HOA 디코딩 이전에 Track 1과 Track 2로부터의 양쪽 HOA 신호들을 혼합할 수 있고, Track 2의 차수를 절삭하여 양쪽 트랙들의 차원을 2D로 줄일 수 있다. 방향 스트림이 존재하는 경우, 이것은 2D HOA로 인코딩된다. 그 다음, 3개 모두의 스트림들이 혼합되어 단일의 HOA 표현을 형성한 다음, 이것이 디코딩되고 재생된다.

[0044] 아래 부분은 도 2에 대응한다. 정면 스테이지용 홀로포닉 시스템과 더 드문드문한 3D 서라운드 시스템을 갖춘 영화관은 전용의 정교한 디코더를 이용하고 스피커 피드들(feeds)을 혼합할 것이다. Track 1 데이터 스트림의 경우, 주변 음향을 나타내는 HOA 데이터는 주변음향 재생 전용의 디코더1로 변환된다. Track 2 데이터 스트림의 경우, HOA(시각적 장면에 관련된 정면 음향)는, 스크린 영역 주변에서 구면 음향 소스의 최상의 배치를 위해 전용 디코더 2에 의해 변환되고 거리 교정(수학식 26)된다. 방향 데이터 스트림은 L개 스피커들에 직접 패닝된다. 3개의 스피커 신호들이 3D 스피커 시스템에 의한 합동 재생을 위해 PCM 혼합된다.

[0045] 고차 앰비소닉을 이용한 음장 표현

[0046] 구면 하모닉스(SH; Spherical Harmonics)를 이용한 음장 표현

[0047] 구면 하모닉스/베셀(Bessel) 표현을 이용할 때, 음향파(acoustic wave) 방정식의 해는 수학식 1에서 제공되는데, M.A. Poletti, "Three-dimensional surround sound systems based on spherical harmonics", Journal of Audio Engineering Society, 53(11), pp.1004-1025, November 2005, and Earl G. Williams, "Fourier Acoustics", Academic Press, 1999를 참조한다.

$$k = \frac{\omega}{c} = \frac{2\pi f}{c}$$

[0048] 음압(sound pressure)은 구면 좌표 r, θ, ϕ (그 정의에 대해 도 7을 참조)와 공간 주파수의 함수이다.

[0049] 이 표현은 관심 또는 유효 영역(도 6에 도시된 바와 같은, 내부 도메인 문제) 바깥의 오디오 음향 소스에 대해 유효하며 직교-정규화된 구면 하모닉스(orthogonal-normalised Spherical Harmonics)를 가정한다:

수학식 1

$$p(r, \theta, \phi, k) = \sum_{n=0}^{\infty} \sum_{m=-n}^n A_n^m(k) j_n(kr) Y_n^m(\theta, \phi)$$

[0050] $A_n^m(k)$ 는 Ambisonic 계수라고 불리며, $j_n(kr)$ 는 제1 종 구면 베셀 함수이고, $Y_n^m(\theta, \phi)$ 는 구면 하모닉스(SH)이며, n은 앰비소닉 차수 인덱스이고, m 은 디그리(degree)를 나타낸다.

kr

[0052] 작은 값(원점(origin)으로부터의 작은 거리 또는 저주파)에 대해서만 유의값(significance value)을 갖는 베셀 함수(Bessel function)의 성질로 인해, 시리즈는 어떤 차수 n에서 중단될 수 있고, 충분한 정확성을 갖는 값 N으로 제약될 수 있다. HOA 데이터 저장시, 대개는 앰비소닉 계수들 A_n^m, B_n^m 또는 일부 미분계수들(상세사항은 이하에서 설명됨)은 차수 N까지 저장된다. N은 앰비소닉 차수라 불린다.

[0053] N 은 앰비소닉 차수라 불리고, 용어 '차수(order)'는 대개는 베셀 함수 $j_n(kr)$ 및 한켈(Hankel) 함수 $h_n(kr)$ 에 서의 n 과 조합하여 사용된다.

[0054] 도 8에 도시된 바와 같이 소스들이 관심 또는 유효 영역 내부에 있는 외부 경우에 대한 파동 방정식의 해는 $r > r_{\text{Source}}$ 에 대해 수학식 2로 표현된다:

수학식 2

$$[0055] p(r, \theta, \phi, k) = \sum_{n=0}^{\infty} \sum_{m=-n}^n B_n^m(k) h_n^{(1)}(kr) Y_n^m(\theta, \phi)$$

[0056] $B_n^m(k)$ 는 다시 한번 앰비소닉 계수라 불리고, $h_n^{(1)}(kr)$ 는 제1종 n 차 구면 한켈 함수를 나타낸다. 공식은 직교-정규화된 SH를 가정한다.

[0057] 주석: 일반적으로, 제1종 구면 한켈 함수 $h_n^{(1)}$ 는 포지티브 주파수에 대한 (e^{ikr} 에 관련된) 송출파(outgoing wave)를 기술하는데 이용되고, 제2종 구면 한켈 함수 $h_n^{(2)}$ 는 (e^{-ikr} 에 관련된) 인입파(incoming wave)에 대해 이용된다. 전술된 "Fourier Acoustics" 서적을 참조.

구면 하모닉스

[0059] 구면 하모닉스 Y_n^m 는 복소수 또는 실수값일 수 있다. HOA에 대한 일반적인 경우는 실수값 구면 하모닉스를 이용한다. 실수 및 복소수 구면 하모닉스를 이용한 앰비소닉의 통일된 표현은, Mark Poletti, "Unified description of Ambisonics using real and complex spherical harmonics", Proceedings of the Ambisonics Symposium 2009, Gras, Austria, June 2009에서 확인될 수 있다.

[0060] (구면 하모닉스가 실수이든 복소수이든지에 관계없이) 구면 하모닉스를 정규화하는 많은 방식들이 있는데, (실수) 구면 하모닉스 및 정규화 방법에 대해 이하의 웹 페이지를 참조한다:

[0061] <http://www.ipgp.fr/~wiecsor/SHTOOLS/www/conventions.html>,

[0062] http://en.citisendum.org/wiki/Spherical_harmonics.

[0063] 정규화는 Y_n^m 와 $Y_{n'}^{m'*}$ 사이의 직교 관계에 대응한다.

$$\int_{S^2} Y_n^m(\Omega) Y_{n'}^{m'}(\Omega)^* d\Omega = \frac{N_{n,m}}{\sqrt{\frac{(2n+1)(n-|m|)!}{4\pi(n+|m|)!}}} \frac{N_{n',m'}}{\sqrt{\frac{(2n'+1)(n'-|m'|)!}{4\pi(n'+|m'|)!}}} \delta_{nn'} \delta_{mm'}$$

[0064] 주석:

[0065] 여기서 S^2 는 단위 구면이고 크로네크 델타(Kroneker delta) $\delta_{aa'}$ 는 $a = a'$ 에 대해서는 1이고, 그 외에는 0이다.

[0066] 복소 구면 하모닉스는 하기와 같이 기술된다:

수학식 3

$$[0067] Y_n^m(\theta, \phi) = s_m \Theta_n^m(\theta) e^{im\phi} = s_m N_{n,m} P_{n,|m|}(\cos(\theta)) e^{im\phi}$$

[0068] 여기서, $i = \sqrt{-1}$ 이고, $s_m = \begin{cases} (-1)^m & m > 0 \\ 1 & \text{else} \end{cases}$ 는 전술된 "Fourier Acoustics" 문헌에서와 같이 양의 m 에 대해 부호가 교호한다. (주석: s_m 은 규약의 항이고, 양의 SH에 대해서만 생략될 수 있다). $N_{n,m}$ 은 직교-정규화된 표현을 위한 형태를 취하는 정규화 항이다(!는 팩토리얼을 나타낸다):

수학식 4

$$[0069] N_{n,m} = \sqrt{\frac{(2n+1)(n-|m|)!}{4\pi (n+|m|)!}}$$

[0070] 이하의 표 1은 복소값 구면 하모닉스에 대한 몇 가지 흔히 이용되는 정규화 방법을 나타낸다. $P_{n,|m|}(x)$ 는 르장드르 함수(Legendre function)와 연관되고, 그 이후에는, 콘돈-肖틀리(Condon-Shortley) 페이즈(phase)라 불리는 페이즈 항 $(-1)^m$ 을 피하고 때때로 다른 표기 내의 P_n^m 표현 내에 포함되는 상기 논문 "Unified description of Ambisonics using real and complex spherical harmonics"로부터의 $|m|$ 을 갖는 표기가 뒤따른다. 연관된 르장드르 함수(Legendre function) $P_{n,|m|} : [-1,1] \rightarrow \mathbb{R}, n \geq |m| \geq 0$ 는 로드리게스 공식(Rodrigues formula)을 이용하여 다음과 같이 표현될 수 있다:

수학식 5

$$[0071] P_{n,|m|}(x) = \frac{1}{2^n n!} (1-x^2)^{\frac{|m|}{2}} \frac{d^{n+|m|}}{dx^{n+|m|}} (x^2 - 1)^n$$

표 1

$N_{n,m}$, 복소 SH에 대한 일반적 정규화 방법			
정규화되지 않음	Schmidt 반-정규화됨, SN3D	4π 정규화됨, N3D, geodesy 4π	Ortho- 정규화됨
1	$\sqrt{\frac{(n- m)!}{(n+ m)!}}$	$\sqrt{\frac{(2n+1)(n- m)!}{(n+ m)!}}$	$\sqrt{\frac{(2n+1)(n- m)!}{4\pi (n+ m)!}}$

[0072]

복소값 구면 하모닉스에 대한 정규화 인자

[0073] 수치적으로, 재귀적 관계(recurrence relationship)로부터 점진적 방식으로 $P_{n,|m|}(x)$ 를 유도하는 것이 유익하며, William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, "Numerical Recipes in C", Cambridge University Press, 1992을 참조한다. $n=4$ 까지의 연관된 르장드르 함수가 표 2에 주어져 있다:

표 2

n	0	1	2	3	4
m					
0	$P_0^0(\cos\theta) = 1$	$P_1^0(\cos\theta) = \cos\theta$	$P_2^0(\cos\theta) = \frac{1}{2}(3\cos^2\theta - 1)$	$P_3^0(\cos\theta) = \frac{1}{2}(5\cos^3\theta - 3\cos\theta)$	$P_4^0(\cos\theta) = \frac{1}{8}(35\cos^4\theta - 30\cos^2\theta + 3)$
1		$P_1^1(\cos\theta) = \sin\theta$	$P_2^1(\cos\theta) = 3\cos\theta\sin\theta$	$P_3^1(\cos\theta) = \frac{3}{2}(5\cos^2\theta - 1)\sin\theta$	$P_4^1(\cos\theta) = \frac{5}{2}(7\cos^3\theta - 3\cos\theta)\sin\theta$
2			$P_2^2(\cos\theta) = 3\sin^2\theta$	$P_3^2(\cos\theta) = 15\cos\theta\sin^2\theta$	$P_4^2(\cos\theta) = \frac{15}{2}(7\cos^2\theta - 1)\sin^2\theta$
3				$P_3^3(\cos\theta) = 15\sin^3\theta$	$P_4^3(\cos\theta) = 105\cos\theta\sin^3\theta$
4					$P_4^4(\cos\theta) = 105\sin^4\theta$

처음 몇 개의 르장드르 다항식

$$P_{n,|m|}(\cos\theta), \quad n=0 \dots 4$$

[0074]

[0075] 실수값 SH는, m 의 반대 값에 대응하는 복소수 컬레 Y_n^m 을 결합함으로써 유도된다(수학식 6의 항 $(-1)^m$ 은 앰비 소닉에서 흔한 경우인 실수 SH에 대한 부호없는 표현을 얻기 위해 도입된다):

수학식 6

$$[0076] S_n^m(\theta, \phi) = \begin{cases} \frac{(-1)^m}{\sqrt{2}}(Y_n^m + Y_n^{m*}) & = \Theta_n^m(\theta) \sqrt{2} \cos(m\phi), \quad m > 0 \\ Y_n^0 & = \Theta_n^0(\theta), \quad m = 0 \\ \frac{(-1)^m}{i\sqrt{2}}(Y_n^{|m|} - Y_n^{|m|*}) & = \Theta_n^{|m|}(\theta) \sqrt{2} \sin(|m|\phi), \quad m < 0 \end{cases}$$

[0077] 이것은, 방위각 항만을 보유하는 $\Phi_m(\phi) = \Phi_{n=|m|}^m(\phi)$ 을 갖는 원형 하모닉스와의 관계를 강조하기 위해 수학식 7과 같이 써어질 수 있다:

수학식 7

$$[0078] S_n^m(\theta, \phi) = \tilde{N}_{n,m} P_{n,|m|}(\cos(\theta)) \Phi_m(\phi)$$

수학식 8

$$[0079] \Phi_{n=|m|}^m(\phi) = \begin{cases} \cos(m\phi), & m > 0 \\ 1 & m = 0 \\ \sin(|m|\phi) & m < 0 \end{cases}$$

[0080] 주어진 앰비소닉 차수 N 에 대해 구면 컴포넌트 S_n^m 의 총 개수는 $(N+1)^2$ 와 같다. 실수값 구면 하모닉스의 일반적 정규화 방식이 표 3에 주어져 있다.

표 3

$\tilde{N}_{n,m}$, 실수 SH에 일반적 정규화 방식			
정규화되지 않음	Schmidt 반-정규화됨, SN3D	4π 정규화됨, N3D, geodesy 4π	Ortho-정규화됨
$\sqrt{2 - \delta_{0,m}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(n - m)!}{(n + m)!}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(2n+1)(n- m)!}{(n+ m)!}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(2n+1)(n- m)!}{4\pi(n+ m)!}}$

3D 실수 SH 정규화방식,
 $\delta_{0,m}$ 은 $m=0$ 에 대해 1의 값을 가지며, 그 외에는 0을 가짐

[0081]

원형 하모닉스

[0083]

2차원 표현을 위해, 하모닉스의 서브셋만이 필요하다. SH 디그리(degree)는 값들 $m \in \{-n, n\}$ 만을 취할 수 있다. 주어진 N에 대한 컴포넌트들의 총 개수는 $2N+1$ 로 줄어드는데, 이것은 기울기 θ 를 나타내는 컴포넌트들이 쓸모없어지고, 구면 하모닉스는 수학식 8에 주어진 원형 하모닉스로 대체될 수 있기 때문이다.

[0084]

3D 앰비소닉 계수를 2D 계수로 변환할 때에 고려될 필요가 있는, 원형 하모닉스에 대한 상이한 정규화 N_m 방식이 존재한다. 원형 하모닉스에 대한 더 일반적인 공식은 하기와 같이 된다:

수학식 9

[0085]

$$\Phi_{n=|m|}^m(\phi) = N_m \Phi_m(\phi) = \begin{cases} N_m \cos(m\phi), & m > 0 \\ N_m, & m = 0 \\ N_m \sin(|m|\phi) & m < 0 \end{cases}$$

[0086]

원형 하모닉스에 대한 몇 가지 일반적인 정규화 인자들이 표 4에 제공되어 있고, 여기서, 정규화 항은 수평항 $\Phi_m(\phi)$ 이전에 인자에 의해 도입된다:

표 4

N_m , 원형 하모닉스에 대한 일반적 정규화 방법			
정규화되지 않음	SN2D	2D 정규화됨, N2D	Ortho- 정규화됨
$\sqrt{\frac{2 - \delta_{0,m}}{2}}$	1	$\sqrt{(2 - \delta_{0,m})}$	$\sqrt{(2 - \delta_{0,m}) \frac{1}{2\pi}}$

2D CH 정규화 방식,

$\delta_{0,m}$ 는 $m=0$ 에 대해서는 1의 값을 갖고, 그 외에는 0을 가짐

[0087]

상이한 정규화들 사이의 변환은 직접적이다. 일반적으로, 정규화는 압력을 기술하는 표기(수학식 1 및 2 참조)와 모든 파생된 고려사항에 영향을 미친다. 정규화의 종류도 역시 앰비소닉 계수에 영향을 미친다. 이들 계수들을 스케일링하기 위해 가중치가 적용될 수 있으며, 예를 들어, Furse-Malham (FuMa) 가중치는 AMB-포맷을 이용하여 파일을 저장할 때 앰비소닉 계수들에 적용된다.

[0089]

2D - 3D 변환에 관련하여, 예를 들어, 2D 확성기 설정을 위해 3D 앰비소닉 표현(레코딩)을 2D 디코더로 디코딩

할 때, CH에서 SH로의 변환 및 그 반대 변환이 앰비소닉 계수들에 적용될 수 있다. 3D-2D 변환의 경우 S_n^m 와 $\Phi_{n=|m|}^m$ 사이의 관계는 앰비소닉 차수 4까지는 이하의 방식으로 기술된다:

		S_0^0						
		S_1^{-1}	S_1^0	S_1^1				
		S_2^{-2}	S_2^{-1}	S_2^0	S_2^1	S_2^2		
		S_3^{-3}	S_3^{-2}	S_3^{-1}	S_3^0	S_3^1	S_3^2	S_3^3
		S_4^{-4}	S_4^{-3}	S_4^{-2}	S_4^{-1}	S_4^0	S_4^1	S_4^2
							S_4^3	S_4^4

[0090]

[0091] 2D에서 3D로의 변환 계수는, $\theta = \frac{\pi}{2}$ 인 수평면에 대해 다음과 같이 유도될 수 있다:

수학식 10

$$\alpha_{\frac{2D}{3D}} = \frac{s_{n=m}^m(\theta=\pi/2, \Phi)}{\Phi_{n=|m|}^m(\phi)} = \frac{\tilde{N}_{m,m} (2m)!}{N_m m! 2^m}$$

[0092]

[0093] 3D로부터 2D로의 변환은 $\frac{1/\alpha_{\frac{2D}{3D}}}{3D}$ 을 이용한다. 상세사항은 아래의 수학식 28, 29, 30과 연계하여 제공된다.

[0094]

직교-정규화로 정규화된 2D 변환은 다음과 같이 된다:

수학식 11

$$\alpha_{\frac{N2D}{ortho3D}} = \sqrt{\frac{(2m+1)!}{4\pi m!^2 2^{2m}}}$$

[0095]

[0096] 앰비소닉 계수

[0097] 앰비소닉 계수들은 음압의 단위 스케일을 가진다: $1Pa = 1 \frac{N}{m^2} = 1 \frac{kg \cdot m}{s^2 \cdot m^2}$. 앰비소닉 계수들은 앰비소닉 신호를 형성하고, 일반적으로 이산 시간의 함수이다. 표 5는, 차원 표시, 앰비소닉 차수 N, 및 앰비소닉 계수들(채널들)의 개수 사이의 관계를 보여준다:

표 5

차원	Ambisonics 계수의 개수			
	Ambisonics 채널의 개수			
	N=1	N=2	N=3	N
	3	5	7	$2 \cdot N+1$
3D	4	9	16	$(N+1)^2$
	Ambisonics 계수의 개수			

[0098]

[0099] 이산 시간 표현을 다룰 때, 대개 앰비소닉 계수들은 다채널 레코딩(채널 = 샘플 v 의 앰비소닉 계수 A_n^m)에 대한 PCM 채널 표현과 같이 인터리빙된 방식으로 저장되며, 계수 시퀀스는 변환 대상이 된다. 3D, N=2의 경우의 예는 다음과 같다:

수학식 12

$$A_0^0(v) \ A_1^{-1}(v) \ A_1^0(v) \ A_1^1(v) \ A_2^{-2}(v) \ A_2^{-1}(v) \ A_2^0(v) \ A_2^1(v) \ A_2^2(v) \ A_0^0(v+1)$$

[0100] 2D, N=2의 경우 예는:

수학식 13

$$A_0^0(v) \ A_1^{-1}(v) \ A_1^1(v) \ A_2^{-2}(v) \ A_2^2(v) \ A_0^0(v+1) \ A_1^{-1}(v+1)$$

[0103] $A_0^0(n)$ 신호는, 방향 정보를 갖지 않지만 레코딩의 일반적 음색 느낌을 나타내는, 앰비소닉 레코딩의 모노 표현으로서 간주될 수 있다.

[0104] 앰비소닉 계수들의 정규화는, 외부 레코딩의 디코딩시에 고려되어야 하는, (이하, 수학식 15를 참조하여 명백해지는 바와 같이) 일반적으로 SH의 정규화에 따라 수행된다(A_n^m 는 정규화 인자 $N_{n,m}$ 을 갖는 SH에 기초하고, \check{A}_n^m 은 정규화 인자 $\check{N}_{n,m}$ 을 갖는 SH에 기초한다):

수학식 14

$$A_n^m = \frac{N_{n,m}}{\check{N}_{n,m}} \check{A}_n^m ,$$

[0105] [0106] 이것은 SN3D 대 N3D 경우에 대해 $A_{N3D,n}^m = \sqrt{(2n+1)} \check{A}_{SN3D,n}^m$ 이 된다.

[0107] B-Format과 AMB 포맷은, 계수에 적용되는 추가 가중치(Gerson, Furse-Malham (FuMa), MaxN 가중치)를 이용한다. 그러면, 기준 정규화는 대개 SN3D이며,

Jérôme Daniel, "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia", PhD thesis, Université Paris 6, 2001, and Dave Malham, "3-D acoustic space and its simulation using ambisonics", http://www.dxarts.washington.edu/courses/567/current/malham_3d.pdf.

[0108] [0109] 를 참조한다.

[0110] 이상적인 평면파 또는 구면파에 대한 파동 방정식의 이하의 2개의 특정 구현은 앰비소닉 계수들에 대한 더 많은 상세사항을 제공한다:

[0111] 평면파

[0112] 평면파 A_n^m 에 대한 과동 방정식의 해를 구하는 것은 k 및 r_s 와는 독립된다; θ_s, ϕ_s 는 소스 각도를 기술하고, '*'은 켤레 복소수를 나타낸다:

수학식 15

$$[0113] A_{n\,plane}^m(\theta_s, \phi_s) = 4\pi i^n P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0114] 여기서 P_{S_0} 은 시간의 함수일 수 있는 기술적 좌표계의 원점에서 측정된 소스의 스케일링 신호 압력을 기술하는데 이용되며, 직교-정규화된 구면 하모닉스에 대하여 $A_0^0_{plane} / \sqrt{4\pi}$ 이 된다.

[0115] 일반적으로, 앰비소닉은 평면파를 가정하고, 앰비소닉 계수

수학식 16

$$[0116] d_n^m(\theta_s, \phi_s) = \frac{A_n^m(\theta_s, \phi_s)}{4\pi i^n} = P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0117] 은 전송되거나 저장된다. 이 가정은 간이 디코더 설계 뿐만 아니라 상이한 방향 신호들의 중첩 가능성을 제공한다. 이것은 1차 B-포맷(N=1)으로 레코딩된 Soundfield™ 마이크로폰의 신호에도 적용되는데, 이것은 등화 필터들의 위상 진행과 비교할 때 명백해진다(이론적 진행의 경우, 앞서 언급된 논문 "Unified description of Ambisonics using real and complex spherical harmonics", chapter 2.1과, 특허보호된 US 4042779호를 참조한다). 수학식 1은 하기와 같이 된다:

수학식 17

$$[0118] p(r, \theta, \phi, k) = \sum_{n=0}^{\infty} \sum_{m=-n}^n j_n(kr) Y_n^m(\theta, \phi) 4\pi i^n P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0119] 계수 d_n^m 은 후처리된 마이크로폰 어레이 신호에 의해 유도되거나, 모노 신호 $P_{S_0}(t)$ 를 이용하여 합성적으로 생성될 수 있는데, 후자의 경우, 방향 구면 하모닉스 $Y_n^m(\theta_s, \phi_s, t)^*$ 는 역시 시간-의존적일 수 있다(움직이는 소스). 수학식 17은 각각의 시간적 샘플링 인스턴스 v 에 대해 유효하다. 합성 인코딩의 프로세스는, 선택된 앰비소닉 차수 N에 대한 벡터/매트릭스 형태로 (매 샘플 인스턴스 v 에 대해) 다시 써어질 수 있다:

수학식 18

$$[0120] \mathbf{d} = \Psi P_{S_0}$$

[0121] \mathbf{d} 는 앰비소닉 신호이고, $d_n^m(\theta_s, \phi_s)$ 를 보유하며, (N=2인 경우의 예):

$$\mathbf{d}(t) = [d_0^0, d_1^{-1}, d_1^0, d_1^1, d_2^{-2}, d_2^{-1}, d_2^0, d_2^1, d_2^2]^T, \text{ size}(\mathbf{d}) = (N+1)^2 \times 1 = 0 \times 1 \text{이고, } P_{S_0} \text{는 기준 원점에서의 소스 신호 압력이고, } \Psi \text{는 인코딩 벡터이며, } Y_n^m(\theta_s, \phi_s)^* \text{를 보유하고, size}(\Psi) = 0 \times 1 \text{이다. (평면파의 방향과)}$$

동일한) 특정 소스 방향 θ_s, ϕ_s 에 대해 구면 하모닉스로부터 인코딩 벡터가 유도될 수 있다.

[0122] 구면파

[0123] $r < r_s$ 에 대해 포인트 소스들(근접장 소스들)에 의해 생성된 인입 구면파를 기술하는 앰비소닉 계수들은 다음과 같다:

수학식 19

$$A_n^m \text{spherical}(k, \theta_s, \phi_s, r_s) = 4\pi \frac{h_n^{(2)}(kr_s)}{h_0^{(2)}(kr_s)} P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0124] [0125] 이 수학식은 아래의 수학식 31 내지 36과 연계하여 유도된다. $P_{S_0} = p(0|r_s)$ 는 원점에서의 음압을 나타내고,

다시 한번 $A_0^0/\sqrt{4\pi}$ 과 같게 되며, $h_n^{(2)}$ 은 제2종 차수 n의 구면 한켈 함수이고, $h_0^{(2)}$ 는 제2종 제로차 구면 한켈 함수이다.

[0126] 수학식 19는, Jérôme Daniel, "Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new ambisonic format", AES 23rd International Conference, Denmark, May 2003에서의 교시와 유사하다. 여기서, 1계 신호에 대한 근접장 효과를 기술하고 있는 M.A. Gerson, "General metatheory of auditory localisation", 92th AES Convention, 1992, Preprint 3306에서,

수학식 11을 기억하면, $\frac{h_n(kr_s)}{h_0(kr_s)} = i^n \sum_{a=0}^n \frac{(n+a)!}{(n-a)!a!} \left(-\frac{ic}{2r_s\omega}\right)^a$, btw $\frac{h_1(kr_s)}{h_0(kr_s)} = i(1 - \frac{ic}{r_s\omega})$ 을 발견할 수 있다.

$$\frac{h_n(kr_s)}{h_0(kr_s)}$$

[0127] 구면 앰비소닉 신호의 합성 생성은 더 높은 앰비소닉 차수 N에 대해서는 덜 일반적인데, 이것은 의 주파수 응답은 저주파의 경우에 수치적으로 다루기 어렵기 때문이다. 이를 수치적 문제는 디코딩/재생을 위한 구면 모델을 이하에서 기술되는 바와 같이 고려함으로써 극복될 수 있다.

[0128] 음장 재생

[0129] 평면파 디코딩

[0130] 일반적으로, 앰비소닉은 원이나 구면 상에 균일하게 분포된 L개의 확성기들에 의한 음장의 재생을 가정한다.

확성기들이 정취자 위치로부터 충분히 멀리 위치해 있다고 가정할 때, 평면파 디코딩 모델은 중심($r_s > \lambda$)에서 유효하다. L개의 확성기에 의해 생성된 음압은 다음과 같이 기술된다:

수학식 20

$$p(r, \theta, \phi, k) = \sum_{n=0}^{\infty} \sum_{m=-n}^n j_n(kr) Y_n^m(\theta, \phi) 4\pi i^n \sum_{l=1}^L w_l Y_n^m(\theta_l, \phi_l)^*$$

[0132] w_l 은 확성기 l 에 대한 신호이고, 음압의 단위 스케일, 1Pa를 가진다. w_l 은 종종 확성기 l 구동 함수라고 불린다. 이 수학식 20 음압은 수학식 17에 의해 기술된 압력과 동일한 것이 바람직하다. 이것은 결과적으로 아래와 같이 된다:

수학식 21

$$\sum_{l=1}^L w_l Y_n^m(\theta_l, \phi_l)^* = d_n^m(\theta_s, \phi_s) = \frac{A_n^m(\theta_s, \phi_s)}{4\pi i^n}$$

[0133]

[0134] 이것은 '재-인코딩 공식' :

수학식 22

$$\mathbf{d} = \Psi \mathbf{y}$$

[0135]

[0136] 여기서, \mathbf{d} 는 앰비소닉 신호이고, $d_n^m(\theta_s, \phi_s)$ 또는 $\frac{A_n^m(\theta_s, \phi_s)}{4\pi i^n}$ 를 보유하며, ($N=2$ 인 경우의 예: $\mathbf{d}_{(n)} = [d_0^0, d_1^{-1}, d_1^0, d_1^1, d_2^{-2}, d_2^{-1}, d_2^0, d_2^1, d_2^2]^T$), $\text{size}(\mathbf{d}) = (N+1)^2 \times 1 = 0x1$ 이고, Ψ 는 (재인코딩) 행렬이며, $Y_n^m(\theta_l, \phi_l)^*$ 를 보유하고, $\text{size}(\Psi) = 0xL$ 이고, \mathbf{Y} 는 확성기 신호 w_l 이며, $\text{size}(\mathbf{Y}_{(n)}, 1) = L$.

[0137]

그러면, \mathbf{Y} 는 한 쌍의 공지된 방법을 이용하여, 예를 들어, 모드 매칭, 또는 특별 스피커 패닝 함수에 최적화된 방법에 의해 유도될 수 있다.

[0138]

구면파 모델에 대한 디코딩

[0139]

더 일반적인 디코딩 모델은 다시 한번, 포인트형 구면파를 방사하는 거리 r_l 을 갖는 원점 주변의 균등 배치된 스피커들을 가정한다. 앰비소닉 계수들 A_n^m 은 수학식 1의 일반적 기술에 의해 주어지며, L 개의 확성기에 의해 생성된 음압은 수학식 19에 따라 주어진다:

수학식 23

$$A_n^m = \sum_{l=1}^L 4\pi \frac{h_n(kr_l)}{h_0(kr_l)} w_l Y_n^m(\theta_l, \phi_l)^*$$

[0140]

[0141] 더욱 정교한 디코더는, $C_n^m = A_n^m \frac{h_0(kr_l)}{4\pi h_n(kr_l)}$ 을 회수한 후 스피커 가중치를 유도하기 위한 $\mathbf{d} = [C_0^0, C_1^{-1}, C_1^0, C_1^1, C_2^{-2}, C_2^{-1}, C_2^0, C_2^1, C_2^2, \dots]$ 과 함께 수학식 17을 적용하기 위해 앰비소닉 계수 A_n^m 을 필터링할 수

있다. 이 모델에 의해 스피커 신호 w_l 은 원점에서의 압력에 의해 결정된다. 상기 언급된 논문 "Three-dimensional surround sound systems based on spherical harmonics"에서 처음 기술된 간단한 소스 접근법을 이용하는 대안적 접근법이 있다. 확성기들은 구면 상에 동일하게 분포되며 2차 소스 특성을 갖는 것으로 가정된다. 이 해법은 Jens Ahrens, Sascha Spors, "Analytical driving functions for higher order ambisonics", Proceedings of the ICASSP, pages 373 - 376, 2008에서 유도되며, 수학식 13은, 앰비소닉 차수

N 에서의 절삭과 확성기 이득 g_l 에 대해 다음과 같이 일반화되어 다시 쓸 수 있다:

수학식 24

$$[0142] w_l = \sum_{n=0}^N \sum_{m=-n}^n g_l \frac{A_n^m}{kr_l h_n^{(2)}(kr_l)} Y_n^m(\theta_l, \phi_l)$$

[0143]

거리 코딩된 앰비소닉 신호

[0144]

기준 스피커 거리 r_{l_ref} 를 이용하여 앰비소닉 인코더에서 C_n^m 을 생성하는 것은, 구면파를 모델링 또는 레코딩할 때 A_n^m 의 수치적 문제를 해결할 수 있다(수학식 18을 이용하여).

수학식 25

$$[0145] C_n^m = A_n^m \frac{h_0(kr_{l_ref})}{4\pi h_n(kr_{l_ref})} = \frac{h_0(kr_{l_ref})}{h_n(kr_{l_ref})} \frac{h_n(kr_s)}{h_0(kr_s)} P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0146]

C_n^m , 기준 거리 r_{l_ref} 및 구면 거리 코딩된 계수들이 이용된다는 표시자가 전송되거나 저장된다.

[0147]

디코더측에서, 수학식 22에서 주어진 간단한 디코딩 처리는 실제 스피커 거리가 $r_l \approx r_{l_ref}$ 인 한 가능하다. 만일 그 차이가 너무 크다면, 앰비소닉 디코딩 이전의 필터링에 의한 수학식 26의 보정이 요구된다.

수학식 26

$$[0148] D_n^m = C_n^m \frac{h_n(kr_{l_ref})}{h_n(kr_l)}$$

[0149]

수학식 24와 같은 다른 디코딩 모델은 결과적으로 거리 코딩된 앰비소닉에 대해 상이한 공식을 유도한다:

수학식 27

$$[0150] \tilde{C}_n^m = \frac{A_n^m}{kr_{l_ref} h_n(kr_{l_ref})} = \frac{1}{kr_{l_ref} h_n(kr_{l_ref})} \frac{h_n(kr_s)}{h_0(kr_s)} P_{S_0} Y_n^m(\theta_s, \phi_s)^*$$

[0151]

또한, 구면 하모닉스의 정규화는 거리 코딩된 앰비소닉의 공식화에 영향을 미칠 수 있다, 즉, 거리 코딩된 앰비소닉 계수는 한정된 정황을 필요로 한다.

[0152]

전술된 2D-3D 변환에 대한 상세사항은 다음과 같다:

[0153]

2D 원형 컴포넌트를 곱셈에 의해 3D 구면 컴포넌트로 변환하는 변환 계수 $\frac{\alpha_{2D}}{3D}$ 는 다음과 같이 유도될 수 있다:

수학식 28

$$\alpha_{\frac{2D}{3D}} = \frac{s_{n=m}^m(\theta=\pi/2, \Phi)}{\Phi_{n=|m|}^m(\phi)} = \frac{\tilde{N}_{m,m} P_{|m|,|m|}(\cos(\theta=\pi/2)) \Phi_m(\phi)}{\mathbf{N}_m \Phi_m(\phi)}$$

[0154]

[0155] 공통 아이덴티티 (2010년 10월 12일자부 Wikipedia, "Associated Legendre polynomials", http://en.wikipedia.org/w/index.php?title=Associated_Legendre_polynomials&oldid=363001511 참조)를 이용하면,

[0156] $P_{l,l}(x) = (2l - 1)!! (1 - x^2)^{l/2}$, 여기서 $(2l - 1)!! = \prod_{i=1}^l (2i - 1)$ 은 이중 팩토리얼이

고, $P_{|m|,|m|}$ 은 다음과 같이 표현될 수 있다:

수학식 29

$$P_{|m|,|m|}(\cos(\theta = \pi/2)) = (2m - 1)!! = \frac{(2m)!}{m! 2^m}$$

[0157]

[0158] 수학식 28에 수학식 29를 대입하면, 수학식 10이 된다.

[0159] 2D로부터 ortho-3D로의 변환은, 관계 $l! = \frac{(l+1)!}{l+1}$ 를 이용하고 $l = 2m$ 를 대입하면, 다음과 같이 유도된다:

수학식 30

$$\alpha_{\frac{N_{2D}}{ortho3D}} = \sqrt{\frac{(2m+1)}{4\pi(2m)!}} \frac{(2m)!}{m! 2^m} = \sqrt{\frac{(2m+1)(2m)!}{4\pi m!^2 2^{2m}}} = \sqrt{\frac{(2m+1)!}{4\pi m!^2 2^{2m}}},$$

[0160]

[0161] 전술된 구면파 확장에 대한 상세사항은 다음과 같다:

[0162] $r < r_s$ 에 대한 포인트 소스와 인입파에 의해 생성된 구면파에 대해 수학식 1을 푸는 것은 더욱 복잡한데, 이것은 소멸성 극미 크기의 포인트 소스는 체적 흐름 Q_s 를 이용하여 기술될 필요가 있기 때문이며, 여기서, r 에서 r_s 의 필드 포인트(field point)에 대한 방출된 압력과 r_s 에 위치한 소스는 다음과 같이 주어진다 (전술된 문헌 "Fourier Acoustics" 참조):

수학식 31

$$p(r|r_s) = -i \rho_0 c k Q_s G(r|r_s)$$

[0163]

[0164] 여기서, ρ_0 은 비-밀도(specific density)이고, $G(r|r_s)$ 은 그린의 함수(Green's function)

수학식 32

$$G(r|r_s) = \frac{e^{-ik|r-r_s|}}{4\pi |r-r_s|}$$

[0165]

[0166] 이며, $G(r|r_s)$ 는 또한, 하기와 같은 수학식에 의해 $r < r_s$ 에 대한 구면 하모닉스로 표현될 수 있다:

수학식 33

$$G(r|r_s) = i k \sum_{n=0}^{\infty} \sum_{m=-n}^n j_n(kr) h_n^{(2)}(kr_s) Y_n^m(\theta, \phi) Y_n^m(\Theta_s, \phi_s)^*$$

[0167]

[0168] 여기서, $h_n^{(2)}$ 은 제2종 한켈 함수이다. 그린의 함수는 단위 meter⁻¹ (k에 기인한 $\frac{1}{m}$) 의 스케일을 가진다는 점에 유의한다. 수학식 31, 33은 하기와 같은 구면파의 앰비소닉 계수를 유도하기 위해 수학식 1과 비교될 수 있다:

수학식 34

$$A_n^m \text{spherical}(k, \Theta_s, \phi_s, r_s) = \rho_0 c k^2 Q_s h_n^{(2)}(kr_s) Y_n^m(\Theta_s, \phi_s)^*$$

[0169]

[0170] 여기서, Q_s 는 단위 m³s⁻¹의 체적 흐름이고, ρ_0 은 kg m⁻³의 비밀도이다.

[0171]

앰비소닉 신호를 합성 생성할 수 있고 상기 평면파 고려사항과 관련시키기 위해, 좌표계의 원점에서 생성된 하기 음압을 이용하여 수학식 34를 표현하는 것은 합리적이다.

수학식 35

$$P_{S_0} = p(0|r_s) = \frac{-i \rho_0 c k Q_s}{4\pi} \frac{e^{-ikr_s}}{r_s} = \frac{\rho_0 c k^2 Q_s}{4\pi} h_0^{(2)}(kr_s)$$

[0172]

[0173] 이것은 결과적으로 하기와 같이 된다.

수학식 36

$$A_n^m \text{spherical}(k, \Theta_s, \phi_s, r_s) = 4\pi \frac{h_n^{(2)}(kr_s)}{h_0^{(2)}(kr_s)} P_{S_0} Y_n^m(\Theta_s, \phi_s)^*$$

[0174]

[0175] 교환 저장 포맷

[0176]

본 발명에 따른 저장 포맷은 하나보다 많은 HOA 표현과 추가의 방향 스트림을 하나의 데이터 컨테이너에 함께 저장하는 것을 허용한다. 이것은 디코더들이 재생을 최적화할 수 있게 하는 상이한 포맷들의 HOA 표현들을 가능케하며, 4GB보다 큰 크기에 대해 효율적인 데이터 저장을 제공한다. 추가의 이점들은 :

[0177]

A) 관련된 저장 포맷 정보와 함께 상이한 포맷을 이용하는 수 개의 HOA 표현의 저장에 의해, 앰비소닉 디코더는

양쪽 표현을 혼합하고 디코드할 수 있다.

- [0178] B) 차세대 HOA 디코더들에 대해 요구되는 정보 항목들이 포맷 정보로서 저장된다:
 - 차원, 관심 영역(청취 영역 외부 또는 내부의 소스), 구면 기초 함수의 정규화;
 - 앰비소닉 계수 팩킹 및 스케일링 정보;
 - 앰비소닉 파동 타입(평면, 구면), (구면파의 디코딩을 위한) 기준 반경;
 - 관련된 방향 모노 신호가 저장될 수 있다. 이들 방향 신호들의 위치 정보가, 각도 및 거리 정보를 이용하거나 앰비소닉 계수들의 인코딩 벡터를 이용하여 기술될 수 있다.
- [0183] C) 앰비소닉 데이터의 저장 포맷이 데이터의 유통성 있고 경제적인 저장을 허용하도록 확장된다:
 - 앰비소닉 컴포넌트(앰비소닉 채널)에 관련된 앰비소닉 데이터를 상이한 PCM-워드 크기 해상도로 저장;
 - 리샘플링 또는 MDCT 처리를 이용하여 감소된 대역폭으로 앰비소닉 데이터를 저장.
- [0186] D) 특별 디코딩(정면, 주변)을 위한 트랙들을 연관시키고, 마이크로폰 신호에 대한 레코딩 정보와 같은 파일에 관한 동반 정보의 저장을 허용하기 위한 메타데이터 필드들이 이용 가능하다:
 - 레코딩 기준 좌표계, 마이크로폰, 소스 및 가상 청취자 위치, 마이크로폰 방향 특성, 룸 및 소스 정보.
- [0188] E) 이 포맷은, 상이한 트랙들을 포함하는 복수의 프레임의 저장에 적합하여, 장면 표현 없이 오디오 장면 변경을 허용한다. (주석: 하나의 트랙은 HOA 음장 표현 또는 단일 소스를 위치 정보와 함께 포함한다. 프레임은 하나 이상의 병렬 트랙들의 조합이다). 트랙들은 프레임의 시작부에서 시작하거나 프레임의 끝에서 끝나므로, 시간 코드가 요구되지 않는다.
- [0189] F) 이 포맷은, 오디오 트랙 데이터의 신속한 액세스(빠른 전진 또는 큐 포인트로의 점프)와 파일 데이터의 시작부의 시간에 관한 시간 코드의 판정을 용이하게 한다.
- [0190] HOA 데이터 교환을 위한 HOA 파라미터
- [0191] 표 6은 HOA 신호 데이터의 명료한 교환을 위해 정의될 필요가 있는 파라미터들을 요약한다. 복소값 및 실수값 경우에 대해 구면 하모닉스의 정의는 고정되며, 수학식 3, 6을 참조한다.

표 6

컨테스트	차원	2D/3D, Ambisonics 계수 (AC)의 팩킹에 영향을 줌
	관심 영역	도6, 도8, 수학식 (1) (2)
	SH 타입	2D의 경우, 복소, 실수 값의, 원형
	SH 정규화	SN3D, N3D, ortho-정규화
Ambisonics- 계수	AC 가중치	B-Format, FuMa, maxN, 가중치 없음, 사용자 정의형
	AC 시퀀스 및 샘플 해상도	수학식 (12) (13)에서의 예, 해상도 16/24 비트 또는 float 타입.
	AC 타입	명시되지 않은 A_n^m , 평면파 타입 d_n^m , 수학식 (16), 거리 코딩된 타입들 D_n^m 또는 \tilde{C}_n^m , 수학식 (26) (27)

HOA 레코딩의 명확한 교환을 위한 파라미터들

[0192]

파일 포맷 상세사항

[0193]

- [0194] 이하에서, HOA(Higher Order Ambisonics) 또는 단일 소스로 구성된 오디오 장면을 위치 정보와 함께 저장하기 위한 파일 포맷이 상세히 기술된다. 오디오 장면은 상이한 정규화 방식들을 이용할 수 있는 복수의 HOA 시퀀스를 포함할 수 있다. 따라서, 디코더는 원하는 확성기 셋업에 대한 대응하는 확성기 신호를, 현재 파일로부터 모든 오디오 트랙의 중첩으로서 계산할 수 있다. 파일은 오디오 콘텐츠를 디코딩하는데 요구되는 모든 데이터를 포함한다. 본 발명에 따른 파일 포맷은 하나보다 많은 HOA 또는 단일 소스 신호를 하나의 파일에 저장하는 특징을 제공한다. 이 파일 포맷은, 각각이 수 개의 트랙을 포함할 수 있는 프레임들의 구성을 이용하며, 여기서 트랙의 데이터는 *TrackPacket*이라 불리는 하나 이상의 패킷에 저장된다.
- [0195] 모든 정수 타입이 리틀-엔디안(little-endian) 바이트 순서로 저장되어 최하위 바이트가 먼저 온다. 비트 순서는 항상 최상위 비트가 먼저 온다. 정수 데이터 타입에 대한 표기는 'int'이다. 선형 'u'는 부호없는 정수를 나타낸다. 비트 해상도는 정의의 끝에 기입된다. 예를 들어, 부호없는 16비트 정수 필드는 'uint16'으로서 정의된다. 정수 포맷의 PCM 샘플 및 HOA 계수들은, 최상위 비트가 소수점인 고정 소수점 숫자로서 표현된다.
- [0196] 모든 부동 소수점 데이터 타입은 IEEE 명세 IEEE-754, "Standard for binary floating-point arithmetic", <http://grouper.ieee.org/groups/754/>를 따른다. 부동 소수점 데이터 타입을 위한 표기는 'float'이다. 비트 해상도는 정의의 끝에 기입된다. 예를 들어, 32 비트 부동 소수점 필드는 'float32'로서 정의된다. 프레임의 시작, 트랙 또는 청크, 및 스트링을 식별하는 상수 식별자 ID는 데이터 타입 바이트로서 정의된다. 바이트 어레이들의 바이트 순서는 최상위 바이트 및 비트 우선이다. 따라서, ID 'TRCK'이 32-비트 바이트 필드에 정의되고, 바이트들은 물리적 순서 'T', 'R', 'C' 및 'K' (<0x54; 0x52; 0x42; 0x4b>)로 기입된다.
- [0197] 16진 값은 '0x'로 시작한다(예를 들어, 0xAB64C5). 단일의 비트들은 인용 부호 내에 놓여지고(예를 들어, '1'), 복수의 2진 값들은 '0b'로 시작한다(예를 들어, 0b0011 = 0x3).
- [0198] 헤더 필드 명칭은 항상, 필드 명칭에 후속하는 필드 명칭으로 시작하고, 각 워드의 첫 문자는 대문자화된다(예를 들어, TrackHeaderSize). 대문자화된 문자들만을 이용하여 필드 또는 헤더 명칭의 약자가 생성된다(예를 들어, TrackHeaderSize = THS).
- [0199] HOA 파일 포맷은 하나보다 많은 Frame, Packet 또는 Track을 포함할 수 있다. 복수의 헤더 필드들의 구분을 위해, 필드 또는 헤더 명칭 다음에 숫자가 올 수 있다. 예를 들어, 세 번째 트랙의 두 번째 TrackPacket은 'Track3Packet2'라고 명명된다. HOA 파일 포맷은 복소값 필드를 포함할 수 있다. 이를 복소값들은 실수부 및 허수부로서 저장되고, 실수부가 먼저 기재된다. 'int8' 포맷의 복소수 1+i2는, '0x01'에 후속하는 '0x02'로서 저장될 것이다. 따라서, 복소값 포맷 타입의 필드 또는 계수는 대응하는 실수값 포맷 타입에 비해 2배의 저장 크기를 요구한다.
- [0200] 고차 앰비소닉 파일 포맷 구조
- [0201] 단일 트랙 포맷
- [0202] 고차 앰비소닉 파일 포맷은, 하나 이상의 Packet에서 하나의 Track을 운반하는 간단한 예시적 HOA 파일 포맷을 보여주는 도 9에 도시된 바와 같이, 적어도 하나의 FileHeader, 하나의 FrameHeader, 하나의 TrackHeader 및 하나의 TrackPacket 을 포함한다.
- [0203] 따라서, HOA 파일의 기본 구조는, 하나의 FileHeader에 후속되는, 적어도 하나의 Track을 포함하는 Frame이다. Track은 항상 TrackHeader와 하나 이상의 TrackPacket들로 구성된다.
- [0204] 복수 프레임 및 트랙 포맷
- [0205] FileHeader와 대조적으로, HOA 파일은 하나보다 많은 Frame을 포함할 수 있고, Frame은 하나보다 많은 Track을 포함할 수 있다. 만일 Frame의 최대 크기가 초과되거나 Track들이 하나의 Frame으로부터 다른 Frame으로 추가되거나 제거된다면, 새로운 FrameHeader가 이용된다. 복수의 Track 및 Frame HOA 파일의 구조가 도 10에 도시되어 있다.
- [0206] 복수의 Track Frame의 구조는, FrameHeader에 후속하는 모든 Frame의 TrackHeader들로 시작한다. 결과적으로, 각 Track의 TrackPacket들은 FrameHeader들에 연속적으로 전송되고, TrackPacket들은 TrackHeader들과 동일한 순서로 인터리빙된다.
- [0207] 복수의 Track Frame에서, 샘플들 내의 Packet의 길이는 FrameHeader에서 정의되고, 모든 Track들에 대해 일정하다.

[0208] 또한, 각 Track의 샘플들은 동기화되고, 예를 들어, Track1Packet1의 샘플들은 Track2Packet1의 샘플들과 동기화된다. 특정 TrackCodingType들은 디코더측에서 지연을 야기할 수 있고, 이러한 특정 지연은 디코더측에 알려질 필요가 있거나, TrackHeader의 TrackCodingType 의존부에 포함되어야 하는데, 이것은 디코더는 모든 TrackPacket을 Frame의 모든 Track의 최대 지연에 동기화하기 때문이다.

[0209] 파일 의존 메타 데이터

[0210] 전체 HOA 파일을 참조하는 메타 데이터는, 선택사항으로서, MetaDataChunks 내의 FileHeader 이후에 추가될 수 있다. MetaDataChunk는 특정 General User ID(GUID)에 후속하는 MetaDataChunkSize로 시작한다. MetaDataChunk의 본질, 예를 들어, Meta Data 정보는 XML 포맷 또는 임의의 사용자-정의형 포맷으로 팩킹된다. 도 11은 수 개의 MetaDataChunk들을 이용한 HOA 파일 포맷의 구조를 도시한다.

[0211] 트랙 타입

[0212] HOA 포맷의 Track은 일반적인 HOATrack과 SingleSourceTrack을 구분한다. HOATrack은 HOACoefficient들로서 코딩된 전체 음장을 포함한다. 따라서, 디코더측에서 계수들을 디코딩하기 위해 장면 표현, 예를 들어, 인코딩된 소스의 위치는 요구되지 않는다. 즉, 오디오 장면은 HOACoefficient 내에 저장된다.

[0213] HOATrack과는 대조적으로, SingleSourceTrack은 오디오 장면 내의 소스의 위치와 함께 PCM 샘플로서 코딩된 단 하나의 소스를 포함한다. 시간의 경과에 따라, SingleSourceTrack의 위치는 고정이거나 가변적일 수 있다. 소스 위치는 TrackHOAEncodingVector 또는 TrackPositionVector로서 전송된다. TrackHOAEncodingVector는 각 샘플에 대해 HOACoefficient를 얻기 위해 HOA 인코딩 값을 포함한다. TrackPositionVector는 소스의 위치를 중앙 정취 위치에 관한 각도와 거리로서 포함한다.

[0214] 파일 헤더

필드 명칭	크기 /비트	데이터 타입	설명
FileID	32	Byte	HOA 파일 포맷에 대한 상수 파일 식별자: <H, "O", "A", "F> or <0x48, 0x4F, 0x41, 0x46>
FileVersionNumber	8	uint8	HOA 포맷의 버전 번호 0-255
FileSampleRate	32	uint32	모든 프레임 및 트랙에 대한 Hz 상수에서의 샘플 레이트
FrameNumberOfFrames	32	uint32	총 프레임 수 적어도 '1'이 요구됨
보류됨	8	byte	
총 비트수	112		

[0215]

[0216] FileHeader는 전체 HOA 파일에 대한 모든 상수(constant) 정보를 포함한다. FileID는 HOA 파일 포맷을 식별하는데 이용된다. 샘플 레이트는 FrameHeader에서 전송되더라도 모든 Track에 대해 일정하다. 프레임별로 샘플 레이트를 변경하는 HOA 파일들은 무효이다. Frame의 개수가 FileHeader에 표시되어 Frame 구조를 디코더에 표시한다.

[0217]

메타 데이터 청크

필드 명칭	크기/비트	데이터 타입	설명
ChunkID	32	byte	GeneralUserID (아직 미정)
ChunkSize	32	uint32	ChunkID 및 the ChunkSize 필드를 제외한 청크의 바이트 단위 크기
ChunkData	8 * ChunkSize	byte	ChunkID에 따라 사용자 정의형 필드 또는 XML-구조
총 비트수	64 + 8 * ChunkSize		

[0218]

[0219] 프레임 헤더

Field Name	Size / Bit	Data Type	Description
FrameID	32	byte	모든 FrameHeader에 대한 상수 식별자: <"F", "R", "A", "M"> or <0x46; 0x52; 0x41; 0x4D>
FrameSize	32	uint32	FrameID 및 FrameSize 필드를 제외한 Frame의 바이트 단위 크기
FrameNumber	32	uint32	첫 번째 Frame에 대해 0으로 시작하고 후속 Frame들에 대해 증가하는 고유 FrameNumber. 마지막 Frame은 FrameNumberOfFileNumber-1을 가짐.
FrameNumberOfSamples	32	uint32	Frame의 각 Track에 저장된 샘플들의 개수
FrameNumberOfTracks	8	uint8	Frame 내에 저장된 Track의 개수
FramePacketSize	32	uint32	샘플에서 Packet의 크기. 패킷 크기는 모든 Track에 대해 일정함.
FrameSampleRate	32	uint32	모든 Frame 및 Track에 대한 Hz 상수에서의 샘플 레이트는 FileSampleRate와 동일해야 함(FileHeader가 알려지지 않을 수 있는 스트리밍 응용에서는 제정의)
Total Number of Bits	200		

[0220]

[0221]

FrameHeader는 Frame의 모든 트랙들의 상수 정보를 유지하고 HOA 파일 내의 변경을 나타낸다. FrameID 및 FrameSize는 Frame의 시작과 Frame의 길이를 나타낸다. 이를 2개 필드는 각 프레임의 용이한 액세스와 Frame 구조의 크로스체크를 허용한다. 만일 Frame 길이가 32비트보다 많은 비트를 요구한다면, 하나의 Frame은 수 개의 Frame들로 분리될 수 있다. 각 Frame은 고유 FrameNumber를 가진다. FrameNumber는 0으로 시작해야 하고, 각 새로운 Frame마다 하나씩 증가되어야 한다.

[0222]

Frame의 샘플들의 개수는 Frame의 모든 Track들에 대해 일정하다. Frame 내의 Track들의 개수는 Frame에 대해 일정하다. 새로운 Frame Header가 원하는 샘플 위치에서 Track을 종료 또는 시작하기 위해 전송된다.

[0223]

각 Track의 샘플들은 Packet에 저장된다. 이를 TrackPacket의 크기는 샘플들에서 표시되고 모든 Track에 대해 일정하다. Packet의 개수는 Frame의 샘플수를 저장하는데 요구되는 정수와 같다. 따라서, Track의 마지막 Packet은 표시된 Packet 크기보다 작은 수의 샘플들을 포함할 수 있다.

[0224]

프레임의 샘플 레이트는 FileSampleRate와 같고 FrameHeader에 표시되어 FileHeader의 지식없이 Frame의 디코딩을 허용한다. 이것은 FileHeader의 지식없이 멀티 프레임 파일의 중간으로부터의 디코딩시, 예를 들어, 스트리밍 애플리케이션에 이용될 수 있다.

[0225]

트랙 헤더

필드 명칭	크기 /비트	데이터 타입	설명
TrackID	32	byte	모든 TrackHeader에 대한 상수 식별자:<T>, <R>, <A>, <C> or <0x54; 0x52; 0x41; 0x43>
TrackNumber	16	uint16	수 개의 Frame 내의 코히어런트 트랙들의 식별을 위한 고유 TrackNumber
TrackHeaderSize	32	uint32	TrackID 및 TrackNumber 필드를 제외한 TrackHeader의 크기 (다음 TrackHeader 또는 첫 번째 TrackPacket의 시작부에 대한 오프셋)
TrackMetaDataOffset	32	uint32	이 필드의 끝으로부터 TrackMetaData 필드의 시작부까지의 오프셋. 제로들은 포함된 TrackMetaData가 없다는 것과 같다
TrackSourceType	1	binary	'0' = HOATrack, 및 '1' = Single sourceTrack
보류됨	7	binary	0b0000000
조건: TrackSourceType == '0'		HOA 트랙들에 대한 TrackHeader	
<HOATrackHeader>	dyn	byte	섹션 HOATrackHeader 참조
조건: TrackSourceType == '1'		SingleSourceTracks에 대한 TrackHeader	
<SingleSourceTrackHeader>	dyn	byte	섹션 단일 소스 고정 위치 트랙 헤더 및 단일 소스 이동 위치 트랙 헤더 참조
조건: TrackMetaDataOffset > 0			
TrackMetaData	dyn	byte	Track 의 존형 metaData를 위한 XML 필드. TrackMetaData table 참조
총 비트수	120 + dyn		

[0226]

[0227]

용어 'dyn'은, 조건적 필드들로 인한 동적 파일 크기를 말한다. TrackHeader는 특정 Track의 Packet에 대한 상수 정보를 보유한다. TrackHeader는 2개의 TrackSourceTypes에 대한 상수와 변수부로 분할된다. TrackHeader는 TrackHeader의 시작부의 검증과 식별을 위한 상수 TrackID로 시작한다. 고유 TrackNumber가 각각의 Track에 할당되어 Frame 보더에 걸친 코히어런트 Track들을 나타낸다. 따라서, 동일한 TrackNumber를 갖는 트랙은 후속 프레임에서 발생할 수 있다. 다음 TrackHeader로의 건너뛰기를 위해 TrackHeaderSize가 제공되고, TrackHeaderSize 필드의 끝으로부터의 오프셋으로서 표시된다. TrackMetaDataOffset는, TrackHeader의 가변 길이부를 건너뛰기 위해 이용될 수 있는 TrackMetaData 필드의 시작부로 바로 점프하기 위한 샘플들의 개수를 제공한다. 제로의 TrackMetaDataOffset은 TrackMetaData 필드가 존재하지 않는다는 것을 나타낸다. TrackSourceType에 따라, HOATrackHeader 또는 SingleSourceTrackHeader가 제공된다. HOATrackHeader는 전체 음장을 기술하는 표준 HOA 계수들에 대한 부가 정보를 제공한다. SingleSourceTrackHeader는 모노 PCM 트랙의 샘플들에 대한 정보와 소스의 위치를 보유한다. SingleSourceTrack들의 경우, 디코더는 Track을 장면 내에 포함시켜야 한다.

[0228]

TrackHeader의 끝에서, 선택사항적인 트랙 의존 Metadata, 예를 들어, A-format 전송을 위한 추가 정보(마이크로폰-어레이 신호)를 제공하기 위해 XML 포맷을 이용하는 선택사항적인 TrackMetaData 필드가 정의된다.

[0229]

HOA 트랙 헤더

필드 명칭	크기 /비트	데이터 타입	설명
TrackComplexValueFlag	2	binary	0b00: 실수부만 0b01: 실수 및 허수부 0b10: 허수부만 0b11: 보류됨
TrackSampleFormat	4	binary	0b0000 부호없는 정수 8비트 0b0001 부호있는 정수 8비트 0b0010 부호있는 정수 16비트 0b0011 부호있는 정수 24비트 0b0100 부호있는 정수 32비트 0b0101 부호있는 정수 64비트 0b0110 Float32비트(2진 단정도) 0b0111 Float64비트(2진 배정도) 0b1000 Float128비트(2진 4배정도) 0b1001-0b1111 보류됨
보류됨	2	binary	비트 채움
TrackHOAParams	dyn	bytes	TrackHOAParams 참조
TrackCodingType	8	uint8	,0' HOA 계수들이 일정한 비트 해상도와 일정한 주파수 해상도를 갖는 PCM 샘플로서 코딩됨. ,1' HOA 계수들이 차수 의존 비트 해상도와 주파수 해상도로 코딩됨 else 추가 코딩 태입을 위해 보류됨

[0230]

조건: TrackCodingType == '1'				코딩 태입 1에 대한 부가 정보
TrackBandwidthReductionType	8	uint8	0 1 2	모든 차수에 대해 전체 대역 폭 MDCT를 이용한 대역 폭 감소 시간 영역 필터를 이용한 대역 폭 감소
TrackNumberOfOrderRegions	8	uint8	대역 폭과 비트 해상도는 다수의 영역들에 대해 적응될 수 있고, 각 수는 시작 및 끝 순서를 가진다. TrackNumberOfOrderRegions는 정의된 영역의 개수를 나타낸다.	

[0231]

각 영역에 대해 이하의 필드들을 기재			
TrackRegionFirstOrder	8	uint8	영역의 처음 차수
TrackRegionLastOrder	8	uint8	이 영역의 마지막 차수

TrackRegionSampleFormat	4	binary	0b0000 부호없는 정수 8비트 0b0001 부호있는 정수 8비트 0b0010 부호있는 정수 16비트 0b0011 부호있는 정수 24비트 0b0100 부호있는 정수 32비트 0b0101 부호있는 정수 64비트 0b0110 Float32비트(2진 단정도) 0b0111 Float64비트(2진 배정도) 0b1000 Float128비트(2진 4배정도) 0b1001-0b1111 보류됨
TrackRegionUseBandwidthReduction	1	binary	'0' 이 영역에 대해 전체 대역 폭 '1' TrackBandwidthReductionType을 갖는 이 영역에 대해 대역 폭을 감소
보류됨	3	binary	비트채움

[0232]

조건: TrackRegionUseBandwidthReduction == '1'			이 영역에서 대역 폭이 감소된다
조건: TrackBandwidthReductionType == 1			MDCT 부가 정보를 이용한 대역 폭 감소
TrackRegionWindowType	8	uint8	0: sine Window: $W(t) = \sin(\frac{\pi(t+0.5)}{N})$ else: 보류됨
TrackRegionFirstBin	16	uint16	첫 번째 코딩된 MDCT 빈(하위 컷-오프 주파수)
TrackRegionLastBin	16	uint16	마지막 코딩된 MDCT 빈(상위 컷-오프 주파수)
조건: TrackBandwidthReductionType == 2			시간 영역 필터 부가 정보를 이용한 대역 폭 감소
TrackRegionFilterLength	16	uint16	저역 통과 필터 계수들의 개수
<TrackRegionFilterCoefficients>	dyn	float32	TrackRegionFilterLength 저역 통과 필터 계수
TrackRegionModulationFreq	32	float32	신호 스펙트럼 시프팅에 요구되는 정규화된 변조 주파수 Ω_{mod}/π
TrackRegionDownsampleFactor	16	uint16	다운샘플링 인자 M은, 젯수 FramePack- et크기이어야 한다
TrackRegionUpsampleFactor	16	uint16	업샘플링 인자 K < M
TrackRegionFilterDelay	16	uint16	인코딩/디코딩 대역 폭 감소 처리의 (FileSampleRate에 따른) 샘플들에서의 지연

[0233]

[0234] HOATrackHeader는 HOATrack을 디코딩하기 위한 정보를 보유하는 TrackHeader의 일부이다. HOATrack의 TrackPacket은 Track의 전체 음장을 코딩하는 HOA 계수들을 전송한다. 기본적으로, HOATrackHeader는 주어진 스피커 셋업에 대해 HOA 계수들을 디코딩하기 위해 디코더측에서 요구되는 모든 HOA 파라미터들을 보유한다.

[0235]

TrackComplexValueFlag 와 TrackSampleFormat은 각 TrackPacket의 HOA 계수들의 포맷 타입을 정의한다. 인코딩되거나 압축된 계수들의 경우, TrackSampleFormat은 디코딩되거나 압축되지 않은 계수들의 포맷을 정의한다. 모든 포맷 타입은 실수 또는 복소수일 수 있다. 상기 섹션 File Format Details에서 복소수에 관한 더 많은 정보가 제공된다.

[0236]

모든 HOA 의존 정보는 TrackHOAParams에서 정의된다. TrackHOAParams는 다른 TrackSourceType들에서 재사용된다. 따라서, TrackHOAParams 필드들은 섹션 TrackHOAParams에서 정의되고 설명된다.

[0237]

TrackCodingType 필드는 HOA 계수들의 코딩(압축) 포맷을 나타낸다. HOA 파일 포맷의 기본 버전은, 예를

들어, 2개의 CodingType들을 포함한다.

[0238] 하나의 CodingType은 PCM 코딩 타입(CodingType == '0')이고, 압축되지 않은 실수 또는 복소수 계수들은 선택된 TrackSampleFormat의 패킷들 내에 기입된다. HOA 계수들의 차수 및 정규화는 TrackHOAParams 필드에 정의된다.

[0239] 제2 CodingType은 샘플 포맷의 변경과 각 HOA 차수의 계수들의 대역폭 제한을 허용한다. 이 CodingType의 상세한 설명은 섹션 TrackRegion Coding에서 제공되고, 간단한 설명은 다음과 같다:

[0240] TrackBandwidthReductionType은 각 HOA 차수의 대역폭을 제한하는데 이용된 처리의 타입을 결정한다. 만일 모든 계수들의 대역폭이 변경되지 않는다면, TrackBandwidthReductionType 필드를 제로로 설정함으로써 대역폭 감소가 스위치 오프될 수 있다. 2개의 다른 대역폭 감소 처리 타입들이 정의된다. 이 포맷은 주파수 영역 MDCT 처리와, 선택사항으로서 시간 영역 필터 처리를 포함한다. MDCT 처리에 관한 더 많은 정보를 위해, 섹션 Bandwidth reduction via MDCT를 참조한다.

[0241] HOA 차수들은 동일한 샘플 포맷과 대역폭의 영역들로 결합될 수 있다. 영역들의 개수는 TrackNumberOfOrderRegions 필드에 의해 표시된다. 각 영역에 대해, 첫 번째 및 마지막 차수 인덱스, 샘플 포맷 및 선택사항적 대역폭 감소 정보가 정의되어야 한다. 영역은 적어도 하나의 차수를 얻을 것이다. 임의의 영역에 의해 커버되지 않는 차수들은 TrackSampleFormat 필드에서 표시된 표준 포맷을 이용하여 전체 대역폭으로 코딩된다. 특별한 경우는 아무 영역도 이용하지 않는 것이다(TrackNumberOfOrderRegions == 0). 이 경우는 PCM 포맷의 디인터리빙된 HOA 계수들에 대해 이용될 수 있고, 여기서, HOA 컴포넌트들은 샘플마다 인터리빙되지 않는다. 영역의 차수들의 HOA 계수들은 TrackRegionSampleFormat으로 코딩된다. TrackRegionUseBandwidthReduction은 영역의 차수들의 계수들에 대해 대역폭 감소 처리의 이용을 나타낸다. 만일 TrackRegionUseBandwidthReduction 플래그가 설정되면, 대역폭 감소 부가 정보가 뒤따를 것이다. MDCT 처리의 경우, 윈도우 타입과 첫 번째 및 마지막 MDCT 빈(bin)이 정의된다. 여기서, 첫 번째 빈은 하위측 컷-오프 주파수와 같고, 마지막 빈은 상위측 컷-오프 주파수와 같다. MDCT 빈은 또한 TrackRegionSampleFormat으로 코딩되며, 섹션 Bandwidth reduction via MDCT를 참조한다.

[0242] 단일 소스 타입

[0243] 단일 소스들은, 고정 위치 소스와 이동 위치 소스로 세분된다. 소스 타입은 TrackMovingSourceFlag에 표시된다. 이동 위치 소스 타입과 고정 위치 소스 타입 사이의 차이는, 고정 소스의 위치는 TrackHeader에서, 이동 소스의 경우에는 각 TrackPackage에서 단 한 번 표시된다는 점이다. 소스의 위치는 구면 좌표계에서 위치 벡터로 명시적으로 표시되거나 HOA 인코딩 벡터로서 목시적으로 표시될 수 있다. 소스 그 자체는, 재생을 위해 앰비소닉 디코더를 이용하는 경우에는 디코더측에서 HOA 계수들로 인코딩되어야 하는 PCM 모노 트랙이다.

[0244]

단일 소스 고정 위치 트랙 헤더

필드 명칭	크기 / 비트	데이터 타입	설명
TrackMovingSourceFlag	1	binary	고정된 소스에 대해 상수 '0'
TrackPositionType	1	binary	'0' 위치가 각도 위치로서 전송 TrackPositionVector[R, theta, phi] '1' 위치가 길이 TrackHOAParamNumberOfCoeffs의 HOA 인코딩 벡터로서 전송
TrackSampleFormat	4	binary	0b0000 부호없는 정수8비트 0b0001 부호있는 정수8비트 0b0010 부호있는 정수16비트 0b0011 부호있는 정수24비트 0b0100 부호있는 정수32비트 0b0101 부호있는 정수64비트 0b0110 Float32비트(2진 단정도) 0b0111 Float64비트(2진 배정도) 0b1000 Float128비트(2진 4배정도) 0b1001-0b1111 보류됨
보류됨	2	binary	비트채움

[0245]

조건: TrackPositionType == '0'			각도 TrackPositionVector로서의 위치
TrackPositionTheta	32	float32	rad [0..pi] 단위의 기울기
TrackPositionPhi	32	float32	rad [0..2pi] 단위의 방위각(반시계 방향)
TrackPositionRadius	32	float32	기준점으로부터 미터 단위 거리
조건: TrackPositionType == '1'			HOA 인코딩 벡터로서의 위치
TrackHOAParams	dyn	bytes	TrackHOAParams 참조
TrackEncodeVectorComplexFlag	2	binary	0b00: 실수부만 0b01: 실수 및 허수부 0b10: 허수부만 0b11: 보류됨 인코딩 벡터에 대해 숫자 타입
TrackEncodeVectorFormat	1	binary	'0' float32 '1' float64
보류됨	5	binary	비트채움

[0246]

조건: TrackEncodeVectorFormat == '0'			float32로서의 인코딩 벡터
<TrackHOAEncodingVector>	dyn	float32	TrackHOAParamCoeffSequence 순서로 된 HOA 인코딩 벡터의 TrackHOAParamNumberOfCoeffs 엔트리들
조건: TrackEncodeVectorFormat == '1'			float64로서의 인코딩 벡터
<TrackHOAEncodingVector>	dyn	float64	TrackHOAParamCoeffSequence 순서로 된 HOA 인코딩 벡터의 TrackHOAParamNumberOfCoeffs 엔트리들

[0247]

고정 위치 소스 타입은 제로의 TrackMovingSourceFlag에 의해 정의된다. 제2 필드는 구면 좌표의 벡터로서 또는 HOA 인코딩 벡터로서 소스 위치의 코딩을 제공하는 TrackPositionType을 나타낸다. 모노 PCM 샘플의 코딩 포맷은 TrackSampleFormat 필드에 의해 표시된다. 만일 소스 위치가 TrackPositionVector로서 전송된다면, 소스 위치의 구면 좌표는, 필드들, TrackPositionTheta (s-축으로부터 x-, y-평면으로의 기울기), TrackPositionPhi (x-축에서 시작하여 반시계방향으로의 방위각) 및 TrackPositionRadius에서 정의된다.

[0248] 만일 소스 위치가 HOA 인코딩 벡터로서 정의된다면, TrackHOAParams 가 먼저 정의된다. 이들 파라미터들은 셱션 TrackHOAParams에 정의되고 HOA 인코딩 벡터의 이용된 정규화와 정의를 나타낸다. TrackEncodeVectorComplexFlag와 TrackEncodeVectorFormat 필드는 후속하는 TrackHOAEncoding 벡터의 포맷 타입을 정의한다. TrackHOAEncodingVector 는 'float32' 또는 'float64' 포맷으로 코딩되는 TrackHOAParamNumberOfCoeffs 값들로 구성된다.

[0249] 단일 소스 이동 위치 트랙 헤더

필드 명칭	크기 / 비트	데이터 타입	설명
TrackMovingSourceFlag	1	binary	이동 소스에 대해서는 상수 '1'
TrackPositionType	1	binary	'0' 위치가 각도 TrackPositionVector[R, theta, phi]으로서 전송 '1' 위치가 길이 TrackHOAParamNumberOfCoeffs의 HOA인코딩 벡터로서 전송
TrackSampleFormat	4	binary	0b0000 부호없는 정수8비트 0b0001 부호있는 정수8비트 0b0010 부호있는 정수16비트 0b0011 부호있는 정수24비트 0b0100 부호있는 정수32비트 0b0101 부호있는 정수64비트 0b0110 Float32비트(2진 단정도) 0b0111 Float64비트(2진 배정도) 0b1000 Float128비트(2진 4배정도) 0b1001-0b1111 보류됨
보류됨	2	binary	비트채움

조건: TrackPositionType == '1'			HOA인코딩 벡터로서의 위치
TrackHOAParams	dyn	bytes	TrackHOAParams 참조
TrackEncodeVectorComplexFlag	2	binary	0b00: 실수부만 0b01: 실수 및 허수부 0b10: 허수부만 0b11: 보류됨 인코딩 벡터에 대해 숫자 타입
TrackEncodeVectorFormat	1	binary	'0' float32 '1' float64
보류됨	5	binary	비트채움

[0250]

[0251] 이동 위치 소스 타입은 '1'의 TrackMovingSourceFlag에 의해 정의된다. 헤더는, 소스 위치 데이터 필드들 TrackPositionTheta, TrackPositionPhi, TrackPositionRadius 및 TrackHOAEncodingVector가 없다는 점을 제외하고는 고정 소스 헤더와 동일하다. 이동 소스의 경우, 이들은 TrackPacket에 위치하여 각 Packet에서 새로운(이동) 소스 위치를 나타낸다.

[0252]

특별 트랙 표

[0253]

TrackHOAParams

필드 명칭	크기 / 비트	데이터 타입	설명
TrackHOAParamDimension	1	binary	'0' = 2D and '1' = 3D
TrackHOAParamRegionOfInterest	1	binary	'0' 관심 영역 외부의 소스에 대해 HOA 계수가 계산되었다 (내부) '1' 관심 영역 내부의 소스에 대해 HOA 계수가 계산되었다 (외부) (관심 영역은 어떠한 소스도 포함하지 않는다.)
TrackHOAParamSphericalHarmonicType	1	binary	'0' 실수 '1' 복소수
TrackHOAParamSphericalHarmonicNorm	3	binary	0b000 경 규화되지 않음 0b001 Schmidt 반-경 규화됨 0b010 4π 경 규화됨 또는 2D 경 규화됨 0b011 Ortho- 경 규화됨 0b100 전용 스케일링 기타 보류됨
TrackHOAParamFurseMalhamFlag	1	binary	HOA 계수들이 Furse-Malham 스케일링에 의해 정 규화됨을 나타낸다

[0254]

필드 명칭	크기 / 비트	데이터 타입	설명
TrackHOAParamDecoderType	2	binary	0b00 평면파 디코더 스케일링: $\frac{1}{(4\pi r^n)}$ 0b01 구면파 디코더 스케일링 (거리 코딩): $\frac{1}{(ikh_n(kr_{ls}))}$ 0b10 구면파 디코더 스케일링 (측정된 음압에 대한 거리 코딩): $h_o(kr_{ls}) / (ikh_n(kr_{ls}))$ 0b11 단순 HOA계수
TrackHOAParamCoeffSequence	2	binary	0b00 B-Format 차수 0b01 수치적 상향 0b10 수치적 하향 0b11 보류됨
보류됨	5	binary	비트채움
TrackHOAParamNumberOfCoeffs	16	uint16	샘플당 HOA 계수들의 개수 - 1
TrackHOAParamHorizontalOrder	8	uint8	X/Y 평면에서 Ambisonics 차수
TrackHOAParamVerticalOrder	8	uint8	3D 차원에 대한 Ambisonics 차수 (2D HOA 계수에 대해서는 '0')

조건: TrackHOAParamSphericalHarmonicNorm == "dedicated" <0b101>	각 HOA 계수에 대한 전용 스케일링 값들을 위한 필드
TrackComplexValueScalingFlag	2 binary 0b00: 실수부만 0b01: 실수 및 허수부 0b10: 허수부만 0b11: 보류됨 전용 TrackScalingValues에 대해 숫자 타입
TrackScalingFormat	1 binary '0': float32 '1': float64
보류됨	5 binary 비트채움

[0255]

조건: TrackScalingFormat='0'	float32로서의 TrackScalingFactors		
<TrackScalingFactors>	dyn	float32	TrackComplexValueScalingFlag == 0b01이면 Track-HOAParamNumberOfCoeffs Scaling Factors, 복소수부의 순서는 <[real1, imaginary1], [real2, imaginary2], ..., [realN, imaginary]>
조건: TrackScalingFormat='1'	Float64로서의 TrackScalingFactors		
<TrackScalingFactors>	dyn	float64	TrackComplexValueScalingFlag == 0b01이면 Track-HOAParamNumberOfCoeffs Scaling Factors, 복소수부의 순서는 <[real1, imaginary1], [real2, imaginary2], ..., [realN, imaginary]>
조건: TrackHOAParamDecoderType == 0b01 TrackHOAParamDecoderType == 0b10	거리 코딩을 위한 기준 확성기 반경이 정의됨		
TrackHOAParamReferenceRadius	16	uint16	이것은 Poletti 또는 Daniel에 따른 구면파 디코더에 대한 HOA 계수들에 적용된 기준 확성기 mm 단위의 반경 r_s 이다.

[0256]

[0257] 과거 HOA 인코딩 및 디코딩을 위한 수 개의 접근법이 논의되어 왔다. 그러나, 코딩 HOA 계수들에 대한 어떠한 결론이나 협의도 없었다. 유익하게도, 본 발명에 따른 포맷은 대부분의 공지된 HOA 표현들의 저장을 허용한다. TrackHOAParams는, 인코더측에서 어떤 종류의 계수들의 정규화와 차수 시퀀스가 사용되었는지를 명확히 하기 위해 정의된다. 이들 정의들은 HOA 트랙들의 혼합과 디코더 매트릭스의 적용을 위해 디코더측에서 고려되어야 한다.

[0258] HOA 계수들은 전체 3차원 음장에 적용되거나, 2차원 x/y-평면에만 적용될 수 있다. HOATrack의 차원은 TrackHOAParamDimension 필드에 의해 정의된다.

[0259] TrackHOAParamRegionOfInterest는, 직렬로 된 2개의 음압 확장을 반영하고, 이에 의해 소스들이 관심 영역의 내부 또는 외부에 주재하고, 관심 영역이 어떠한 소스도 포함하지 않는다. 내부 및 외부 경우에 대한 음압의 계산은 상기 수학식 1 및 2에 각각 정의되고, 이로써 HOA 신호 $A_n^m(k)$ 의 방향 정보가 결례 복소 구면 하모닉 함수 $Y_n^m(\theta, \phi)^*$ 에 의해 결정된다. 이 함수는 복소수 및 실수 버전으로 정의된다. 인코더와 디코더는 등가 숫자 타입의 구면 하모닉 함수를 적용해야 한다. 따라서, TrackHOAParamSphericalHarmonicType은, 어떤 종류의 구면 하모닉 함수가 인코더측에서 적용되었는지를 나타낸다.

[0260] 전술된 바와 같이, 기본적으로 구면 하모닉 함수는 연관된 르장드르 함수와 복소 또는 실수 삼각 함수에 의해 정의된다. 연관된 르장드르 함수는 수학식 5에 의해 정의된다. 복소값 구면 하모닉 표현은 하기와 같고,

$$Y_n^m(\theta, \phi) = N_{n,m} P_{n,|m|}(\cos(\theta)) e^{im\phi} \begin{cases} (-1)^m & ; m \geq 0 \\ 1 & ; m < 0 \end{cases}$$

[0261]

[0262] 여기서, $N_{n,m}$ 은 스케일링 계수이다(수학식 3 참조). 이 복소값 표현은 다음과 같은 수학식을 이용하여 실수값 표현으로 변환될 수 있다:

$$S_n^m(\theta, \phi) = \begin{cases} \frac{(-1)^m}{\sqrt{2}} (Y_n^m + Y_n^{m*}) & = \tilde{N}_{n,m} P_{n,|m|}(\cos(\theta)) \cos(m\phi), & m > 0 \\ Y_n^0 & = \tilde{N}_{n,m} P_{n,|m|}(\cos(\theta)) & m = 0. \\ \frac{-1}{i\sqrt{2}} (Y_n^m - Y_n^{m*}) & = \tilde{N}_{n,m} P_{n,|m|}(\cos(\theta)) \sin(|m|\phi), & m < 0 \end{cases}$$

[0263]

[0264] 여기서, 실수값 구면 하모닉스에 대한 수정된 스케일링 계수는 $\tilde{N}_{n,m} = \sqrt{2 - \delta_{0,m}} N_{n,m}$, $\delta_{0,m} = \begin{cases} 1 & ; m = 0 \\ 0 & ; m \neq 0 \end{cases}$ 이다.

[0265] 2D 표현의 경우, 원형 하모닉스 함수가 HOA 계수들의 인코딩 및 디코딩에 이용되어야 한다. 원형 하모닉스의 복소값 표현은 $\tilde{Y}_m(\phi) = \tilde{N}_m e^{im\phi}$ 에 의해 정의된다. 원형 하모닉스의 실수값 표현은 $\check{S}_m(\phi) = \tilde{N}_m \begin{cases} \cos(m\phi) & ; m \geq 0 \\ \sin(|m|\phi) & ; m < 0 \end{cases}$ 에 의해 정의된다.

[0266] 구면 또는 원형 하모닉 함수를 특정 응용이나 요건에 맞추기 위해 수 개의 정규화 인자들 $N_{n,m}$, $\tilde{N}_{n,m}$, \tilde{N}_m 및 $\tilde{\tilde{N}}_m$ 이 이용된다. HOA 계수들의 올바른 디코딩을 보장하기 위해, 인코더측에서 이용된 구면 하모닉 함수의 정규화가 디코더측에 알려져야 한다. 이하의 표 7은 TrackHOAParamSphericalHarmonicNorm 필드로 선택될 수 있는 정규화를 정의한다.

표 7

3D 복소값 구면 하모닉 정규화 $N_{n,m}$			
정규화되지 않음 0b000	Schmidt 반 정규화됨, SN3D 0b001	4π 정규화됨, N3D, Geodesy 4π 0b010	Ortho- 정규화됨 0b011
1	$\sqrt{\frac{(n- m)!}{(n+ m)!}}$	$\sqrt{\frac{(2n+1)(n- m)!}{(n+ m)!}}$	$\sqrt{\frac{(2n+1)(n- m)!}{4\pi(n+ m)!}}$
3D 실수값 구면 하모닉 정규화 $\tilde{N}_{n,m}$			
정규화되지 않음 0b000	Schmidt 반 정규화됨, SN3D 0b001	4π 정규화됨, N3D, Geodesy 4π 0b010	Ortho- 정규화됨 0b011
$\sqrt{2 - \delta_{0,m}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(n- m)!}{(n+ m)!}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(2n+1)(n- m)!}{(n+ m)!}}$	$\sqrt{(2 - \delta_{0,m}) \frac{(2n+1)(n- m)!}{4\pi(n+ m)!}}$
2D 복소값 원형 하모닉 정규화 \tilde{N}_m			
정규화되지 않음 0b000	Schmidt 반 정규화됨, SN2D 0b001	2D 정규화됨, N2D, 0b010	Ortho- 정규화됨 0b011
$\sqrt{\frac{1}{2}}$	$\sqrt{\frac{1 + \delta_{0,m}}{2}}$	1	$\sqrt{\frac{1}{2\pi}}$
2D 실수값 원형 하모닉 정규화 $\tilde{\tilde{N}}_m$			
정규화되지 않음 0b000	Schmidt 반 정규화됨, SN2D 0b001	2D 정규화됨, N2D, 0b010	Ortho- 정규화됨 0b011
$\sqrt{\frac{2 - \delta_{0,m}}{2}}$	1	$\sqrt{(2 - \delta_{0,m})}$	$\sqrt{(2 - \delta_{0,m}) \frac{1}{2\pi}}$

구면 및 원형 하모닉 함수들의 정규화

[0267]

[0268] 향후의 정규화를 위해, TrackHOAParamSphericalHarmonicNorm 필드의 전용값이 이용가능하다. 전용 정규화를 위해, 각 HOA 계수의 스케일링 계수가 TrackHOAParams의 끝에서 정의된다. 전용 스케일링 계수 TrackScalingFactors는 실수 또는 복소 'float32' 또는 'float64' 값으로서 전송될 수 있다. 전용 스케일링의 경우 스케일링 계수 포맷은, TrackComplexValueScalingFlag 및 TrackScalingFormat 필드들에 정의된다.

[0269]

Furse-Malham 정규화는, 상이한 HOA 차수들의 계수들의 진폭을, 정수 포맷 타입의 전송에 대해 '1'보다 작은 절대값으로 등화하기 위해 코딩된 HOA 계수들에 추가로 적용될 수 있다. Furse-Malham 정규화는, 차수 3 계수들 까지의 SN3D 실수값 구면 하모닉 함수에 대해 설계되었다. 따라서, SN3D 실수값 구면 하모닉 함수와의 조합으로만 Furse-Malham 정규화를 이용하는 것이 추천된다. 게다가, 3보다 큰 HOA 차수를 갖는 Track들에 대해서는 TrackHOAParamFurseMalhamFlag가 무시된다. Furse-Malham 정규화는 HOA 계수들을 디코딩하기 위해 디코더측에서 반전(invert)되어야 한다. 표 8은 Furse-Malham 계수들을 정의한다.

표 8

n	m	Furse-Malham 가중치
0	0	$1/\sqrt{2}$
1	-1	1
1	0	1
1	1	1
2	-2	$2/\sqrt{3}$
2	-1	$2/\sqrt{3}$
2	0	1
2	1	$2/\sqrt{3}$
2	2	$2/\sqrt{3}$
3	-3	$\sqrt{8/5}$
3	-2	$3/\sqrt{5}$
3	-1	$\sqrt{45/32}$
3	0	1
3	1	$\sqrt{45/32}$
3	2	$3/\sqrt{5}$
3	3	$\sqrt{8/5}$

인코더측에서 적용될 Furse-Malham 정규화 계수들

[0270]

[0271] `TrackHOAParamDecoderType`은 어떤 종류의 디코더가 디코더측에 존재한다고 인코더측이 가정하는지를 정의한다. 디코더 타입은, 음장을 렌더링하기 위해 디코더측에서 사용될 확성기 모델(구면 또는 평면파)을 결정한다. 이로써, 디코더 방정식의 부분들을 인코더 방정식으로 이동시킴으로써 디코더의 계산 복잡성이 줄어들 수 있다. 추가로, 인코더측에서의 수치 문제가 줄어들 수 있다. 또한, 디코더측에서의 모든 불일치는 인코더로 이전될 수 있기 때문에, 디코더는 모든 HOA 계수들에 대해 동일한 처리로 줄어들 수 있다. 그러나, 구면파의 경우, 청취 위치로부터 확성기들의 일정한 거리가 가정되어야 한다. 따라서, 가정된 디코더 타입이 `TrackHeader`에 표시

되고, 구면파 디코더 타입에 대한 확성기 반경 r_{ls} 은 선택사항적 필드인 밀리미터 단위의 `TrackHOAParamReferenceRadius`에서 전송된다. 디코더측의 추가 필터는 가정된 확성기 반경과 실제 확성기 반경

사이의 차이를 등화할 수 있다. HOA 계수들의 `TrackHOAParamDecoderType` 정규화는 `TrackHOAParamRegionOfInterest`에서 선택된 내부 또는 외부의 직렬로 된 음장 확장의 이용에 의존한다. 주석:

수학식 18 및 이하의 수학식들에서의 계수들 d_n^m 은 하기에서 계수들 C_n^m 에 대응한다. 인코더측에서 계수들

C_n^m 은 표 9에 정의된 계수들 A_n^m 또는 B_n^m 로부터 결정되고, 저장된다. 사용되는 정규화는 `TrackHOAParam` 헤더의 `TrackHOAParamDecoderType` 필드에서 표시된다:

표 9

<i>TrackHOAParamDecoderType</i>	HOA 계수 내부	HOA 계수 외부
0b00: 평면파	$C_n^m = A_n^m / (4\pi i^n)$	---
0b01: 구면파	$C_n^m = A_n^m / (ikh_n(kr_{ls}))$	$C_n^m = A_n^m / (ikj_n(kr_{ls}))$
0b10: 구면파 측정된 음장	$C_n^m = A_n^m h_0(kr_{ls}) / (h_n(kr_{ls}))$	$C_n^m = A_n^m h_0(kr_{ls}) / (j_n(kr_{ls}))$
0b11: 정규화되지 않음	$C_n^m = A_n^m$	$C_n^m = B_n^m$

[0272] 수 개의 디코더 타입에 대한 전송된 HOA 계수들의 정규화

[0273] 하나의 시간 샘플의 HOA 계수들은 *TrackHOAParamNumberOfCoeffs(0)* 개의 계수들 C_n^m 을 포함한다. N 은 HOA 계수들의 차원에 의존한다. 2D의 경우 음장 O 는 $2N + 1$ 와 같고, 여기서 N 은 *TrackHOAParam* 헤더로부터의 *TrackHOAParamHorizontalOrder* 필드와 같다. 2D HOA 계수들은 $-N \leq m \leq N$ 인 $C_{|m|}^m = C_m$ 로서 정의되고, 표 10에 도시된 3D 계수들의 서브셋으로서 표현될 수 있다.

[0274] 3D의 경우 음장 O 는 $(N + 1)^2$ 와 같고, 여기서 N 은 *TrackHOAParam* 헤더로부터의 *TrackHOAParamVerticalOrder* 필드와 같다. 3D HOA 계수들은 C_n^m 은 $0 \leq n \leq N$ 및 $-n \leq m \leq n$ 에 대해 정의된다. HOA 계수들의 일반적 표현이 표 10에 주어져 있다:

표 10

		C_0^0					
	C_1^{-1}	C_1^0	C_1^1				
	C_2^{-2}	C_2^{-1}	C_2^0	C_2^1	C_2^2		
	C_3^{-3}	C_3^{-2}	C_3^{-1}	C_3^0	C_3^1	C_3^2	C_3^3
C_4^{-4}	C_4^{-3}	C_4^{-2}	C_4^{-1}	C_4^0	C_4^1	C_4^2	C_4^3

[0275] 3D 계수의 부분집합으로서 볼드체로 2D 계수를 도시하는 4차까지의 HOA 계수들의 표현

[0276] 3D 음장과 *TrackHOAParamVerticalOrder* 보다 큰*TrackHOAParamHorizontalOrder*의 경우, 혼합-차수 디코딩이 수행될 것이다. 혼합-차수-신호에서, 일부 고차 계수들은 2D에서만 전송된다. *TrackHOAParamVerticalOrder* 필드는 모든 계수들이 전송되는 수직 차수를 결정한다. 수직 차수로부터 *TrackHOAParamHorizontalOrder*까지는 2D 계수들만이 사용된다. 따라서, *TrackHOAParamHorizontalOrder*는 *TrackHOAParamVerticalOrder*보다 크거나 같다. 수평 차수 4와 수직 차수 2의 혼합-차수 표현의 예가 표 11에 도시되어 있다.

표 11

			C_0^0	
	C_1^{-1}	C_1^0	C_1^1	
C_2^{-2}	C_2^{-1}	C_2^0	C_2^1	C_2^2
				C_3^3
C_3^{-3}				C_4^4
C_4^{-4}				

수직 차수 2와 수평 차수 4의 혼합-차수 표현에 대한 HOA 계수들의 표현

[0277]

[0278] HOA 계수들 C_n^m 은 Track의 Packet들에 저장된다. 계수들의 시퀀스, 예를 들어, 어느 계수가 먼저 오고 어느 것이 그 뒤를 따르는지가, 과거에는 상이하게 정의되었다. 따라서, 필드 TrackHOAParamCoeffSequence는 계수 시퀀스들의 3가지 탑입을 나타낸다. 표 10의 HOA 계수 배열로부터 3개의 시퀀스가 유도된다.

[0279] B-Format 시퀀스는 표 12에 도시된 바와 같이 차수 3까지의 HOA 계수들에 대해 특별한 워딩(wording)을 사용한다:

표 12

			W		
	Y	S	X		
V	T	R	S	U	
Q	O	M	K	L	N P

B-Format HOA 계수 명명 규약

[0280]

[0281] B-Format의 경우 HOA 계수들은 최하 차수로부터 최고 차수로 전송되며, 각 차수의 HOA 계수들은 알파벳 순서로 전송된다. 예를 들어, HOA 차수 3의 3D 셋업의 계수들은 시퀀스 W, X, Y, S, R, S, T, U, V, K, L, M, N, O, P 및 Q로 저장된다. B-포맷은 세 번째 HOA 차수까지만 정의된다. 수평 (2D) 계수들의 전송의 경우, 보충 3D

계수들은 무시된다, 예를 들어, W, X, Y, U, V, P, Q. 3D HOA에 대한 계수들 C_n^m 은 최하 HOA 차수로부터 최상 HOA 차수까지 수치적으로 상향 또는 하향 방식으로 TrackHOAParamCoeffSequence로 전송된다($n = 0 \dots N$).

수치적 상향 시퀀스는 $m = -n$ 과 함께 시작하고 $m = n$ 까지 증가하는데, 이것은 Chris Travis, "Four candidate component sequences", <http://ambisonics.googlegroups.com/web/Four+candidate+component+sequences+V09.pdf>, 2008에서 정의된

'CG' 시퀀스이다. 수치적 하향 시퀀스 $m = n$ 은 $m = -n$ 으로부터 $C_0^0, C_1^1, C_1^0, C_1^{-1}, C_2^2, C_2^1, C_2^0, C_2^{-1}, C_2^{-2}, \dots$ 까지 다른 방식으로 주행하고, 이것은 상기 개시물에서 정의된 'QM' 시퀀스이다.

[0282] 2D HOA 계수의 경우, TrackHOAParamCoeffSequence 수치적 상향 및 하향 시퀀스들은 3D 경우에서와 같지만,

$|m| \neq n$ 인 미사용 계수들(즉, 표 10에서 부채꼴 HOA 계수들 $C_{|m|}^m = C_m$)은 생략된다. 따라서, 수치적 상향

$C_0^0, C_1^{-1}, C_1^1, C_2^{-2}, C_2^2, \dots)$ 시퀀스는 $C_0^0, C_1^1, C_1^{-1}, C_2^2, C_2^{-2}, \dots)$ 로 이어지고, 수치적 하향 시퀀스는 $C_0^0, C_1^1, C_1^{-1}, C_2^2, C_2^{-2}, \dots)$ 로 이어진다.

[0283] 트랙 패킷

[0284] HOA 트랙 패킷

[0285] PCM 코딩 타입 패킷

필드 명칭	크기 / 비트	데이터 타입	설명
<PacketHOACoeffs>	dyn	dyn	TrackSampleFormat과 TrackHOAParamCoeffSequence, 예를 들어 <[W(0), X(0), Y(0), S(0)], [W(1), X(1), Y(1), S(1)], ..., S(FrameNumberOfSamples-1)]>로 저장된 채널 인터리빙된 HOA 계수

[0286]

[0287] 이 Packet은 TrackHOAParamCoeffSequence에서 정의된 순서로 HOA 계수들 C_n^m 을 포함하며, 하나의 시간 샘플의 모든 계수들은 연속 전송된다. 이 Packet은, 제로의 TrackSourceType과 제로의 TrackCodingType을 갖는 표준 HOA 트랙들에 대해 이용된다.

[0288] 동적 해상도 코딩 타입 패킷

필드 명칭	크기 / 비트	데이터 타입	설명
<PacketHOACoeffsCoded>	dyn	dyn	TrackCodingType, e.g. <[W(0), W(1), W(2), ...], [X(0), X(1), X(2), ...], [Y(0), Y(1), Y(2), ...], [S(0), S(1), S(2), ...]>에 따라 저장된 채널 디-인터리빙된 HOA 계수

[0289]

[0290] 'zero'의 TrackSourceType과 '1'의 TrackCodingType에 대해 동적 해상도 팩키지가 이용된다. TrackOrderRegions의 상이한 해상도는 각 TrackOrderRegion에 대해 상이한 저장 크기로 이어진다. 따라서, HOA 계수들은 디인터리빙된 방식으로 저장되고, 예를 들어, 하나의 HOA 차수의 모든 계수들은 연속 저장된다.

[0291] 단일 소스 트랙 패킷

[0292] 단일 소스 고정 위치 패킷

필드 명칭	크기 / 비트	데이터 타입	설명
<PacketMonoPCMTrack>	dyn	dyn	TrackSampleFormat으로 저장된 단일 오디오 소스의 PCM 샘플들

[0293]

[0294] 단일 소스 고정 위치 패킷은 '1'의 TrackSourceType과 '제로'의 TrackMovingSourceFlag에 이용된다. Packet은 모노 소스의 PCM 샘플들을 보유한다.

[0295] 단일 소스 이동 위치 패킷

필드 명칭	크기 / 비트	데이터 타입	설명
PacketDirectionFlag	1	binary	방향이 변경되면 '1'로 설정. 프레임의 첫 번째 Packet에 대해서는 '1'이 강제적임
보류됨	7	binary	비트채움

조건: <code>PacketDirectionFlag == '1'</code>			새로운 위치 데이터
조건: <code>TrackPositionType == '0'</code>			각도 TrackPositionVector로서의 위치 TrackPositionVector
theta	32	float32	rad [0..pi] 단위의 기울기
phi	32	float32	rad [0..2pi] 단위의 방위 각(반시계 방향)
radius	32	float32	기준점으로부터의 미터 단위 거리

조건: <code>TrackPositionType == '1'</code>			HOA 인코딩 벡로서의 위치
조건: <code>TrackEncodeVectorFormat == '0'</code>			float32로서의 인코딩 벡터
<TrackHOAEncodingVector>	dyn	float32	TrackHOAParamCoeffSequence 순서로 된 HOA 인코딩 벡터의 TrackHOAParamNumberOfCoeffs 엔트리들
조건: <code>TrackEncodeVectorFormat == '1'</code>			float64로서의 인코딩 벡터
<TrackHOAEncodingVector>	dyn	float64	TrackHOAParamCoeffSequence 순서로 된 HOA 인코딩 벡터의 TrackHOAParamNumberOfCoeffs 엔트리들

<PacketMonoPCMTrack>		dyn	dyn	TrackSampleFormat으로 저장된 단일 오디오 소스의 PCM 샘플들

[0296]

[0297] 단일 소스 이동 위치 패킷은 '1'의 TrackSourceType과 '1'의 TrackMovingSourceFlag에 이용된다. 이것은 TrackPacket의 모노 PCM 샘플들과 샘플에 대한 위치 정보를 보유한다.

[0298]

PacketDirectionFlag는 Packet의 방향이 변경되었는지 또는 이전 Packet의 방향이 이용되어야 하는지를 나타낸다. 각 Frame의 시작부로부터의 디코딩을 보장하기 위해, PacketDirectionFlag는 Frame의 첫 번째 이동 소스 TrackPacket에 대해 '1'과 같다.

[0299]

'1'의 PacketDirectionFlag의 경우, 후속하는 PCM 샘플 소스의 방향 정보가 전송된다. TrackPositionType에 따라, 방향 정보는 구면 좌표의 TrackPositionVector로서 전송되거나 정의된 TrackEncodingVectorFormat을 갖는 TrackHOAEncodingVector로서 전송된다. TrackEncodingVector는 HOAParamHeader 필드 정의에 순응하는 HOA 계수들을 생성한다. 방향 정보에 연속하여, TrackPacket의 PCM 모노 샘플들이 전송된다.

[0300]

코딩 처리

[0301]

트랙영역 코딩

[0302]

HOA 신호들은 마이크로폰 어레이를 이용한 음장 레코딩으로부터 유도될 수 있다. 예를 들어, WO 03/061336 A1에 개시된 Eigenmike는 차수 3의 HOA 레코딩을 얻는데 이용될 수 있다. 그러나, 마이크로폰 어레이의 유한 크기는 레코딩된 HOA 계수들에 대한 제약으로 이어진다. WO 03/061336 A1에서, 그리고 전술된 논문 "Three-dimensional surround sound systems based on spherical harmonics"에서, 유한 마이크로폰 어레이에 의해 야기되는 문제가 논의된다.

[0303]

마이크로폰 캡슐의 거리는 결과적으로 공간 샘플링 이론에 의해 주어지는 상위 주파수 한계를 초래한다. 이 상위 주파수 위에서 마이크로폰은 올바른 HOA 계수들을 생성할 수 없다. 또한, HOA 청취 위치로부터 마이크로폰

의 유한 거리는 등화 필터를 요구한다. 이들 필터들은 심지어 각 HOA 차수에 따라 증가하는 저주파에 대해 높은 이득을 얻는다. WO 03/061336 A1에서, 등화 필터의 동적 범위를 다루기 위하여 고차 계수들에 대한 하위 커트오프 주파수가 도입된다. 이것은, 상이한 HOA 차수들의 HOA 계수들의 대역폭은 상이할 수 있다는 것을 보여준다. 따라서, HOA 파일 포맷은 각 HOA 차수에 대한 필요한 주파수 대역폭만의 송신을 가능케하는 TrackRegionBandwidthReduction 을 제공한다.

[0304] 등화 필터의 높은 동적 범위로 인해, 그리고 제로 차수 계수는 기본적으로 모든 마이크로폰 신호들의 합이라는 사실로 인해, 상이한 HOA 차수들의 계수들은 상이한 동적 범위를 가질 수 있다. 따라서, HOA 파일 포맷은, 포맷 타입을 각 HOA 차수의 동적 범위에 맞게 하기 위한 특징도 역시 제공한다.

트랙영역 인코딩 처리

[0306] 도 12에 도시된 바와 같이, 인터리빙된 HOA 계수들은, 첫 번째 TrackRegion에 할당된 첫 번째 디-인터리빙 단계 또는 스테이지(1211)에 공급되고, TrackRegion의 모든 HOA 계수들을 FramePacketSize 샘플들에 대한 디-인터리빙된 베퍼들로 분리한다. TrackRegion의 계수들은 HOA Track Header의 TrackRegionLastOrder 및

TrackRegionFirstOrder 필드로부터 유도된다. 디-인터리빙이란, n 과 m 의 한 조합에 대한 계수들 C_n^m 이 하나의 베퍼 내에 그룹화된다는 것을 의미한다. 디인터리빙 단계 또는 스테이지(1211)로부터, 디인터리빙된 HOA 계수들은 TrackRegion 인코딩 셱션으로 전달된다. 잔여 인터리빙된 HOA 계수들은, 디-인터리빙 단계 또는 스테이지(121N)까지 후속하는 TrackRegion 디-인터리빙 단계 또는 스테이지 등으로 전달된다. 디-인터리빙 단계 또는 스테이지의 개수 N은 TrackNumberofOrderRegions + '1'과 같다. 추가의 디-인터리빙 단계 또는 스테이지(125)는 TrackRegion의 일부가 아닌 잔여 계수들을 디인터리빙하여 포맷 변환 단계 또는 스테이지(126)를 포함한 표준 처리 경로에 둔다.

[0307] TrackRegion 인코딩 경로는 선택사항적인 대역폭 감소 단계 또는 스테이지(1221)와 변환 단계 또는 스테이지(1231)를 포함하고, 각 HOA 계수 베퍼에 대한 병렬 처리를 수행한다. 대역폭 감소는 TrackRegionUseBandwidthReduction 필드가 '1'로 설정되면 수행된다. 선택된 TrackBandwidthReductionType에 따라, HOA 계수들의 주파수 범위를 제한하고 이들을 임계적으로 다운샘플링하기 위한 처리가 수행된다. 이것은 HOA 계수들의 개수를 샘플들의 최소 요구 개수로 줄이기 위해 수행된다. 포맷 변환은 현재의 HOA 계수 포맷을 HOATrack header에 정의된 TrackRegionSampleFormat으로 변환한다. 이것은, HOA 계수들을 HOA Track Header의 표시된 TrackSampleFormat으로 변환하는 표준 처리 경로에서의 유일한 단계/스테이지이다. 멀티플렉서 TrackPacket 단계 또는 스테이지(124)는, HOA 계수 베퍼들을, 선택된 TrackHOAParamCoeffSequence 필드에서 정

의된 TrackPacket 데이터 파일 스트림으로 멀티플렉싱하며, 여기서, n 과 m 인덱스의 한 조합에 대한 계수 C_n^m 은 (하나의 베퍼 내에서) 디-인터리빙되어 유지된다.

트랙영역 디코딩 처리

[0309] 도 13에 도시된 바와 같이, 디코딩 처리는 인코딩 처리와는 반대이다. 디멀티플렉서 단계 또는 스테이지(134)는, 표시된 TrackHOAParamCoeffSequence로부터의 TrackPacket 데이터 파일 또는 스트림을 디인터리빙된 HOA 계수 베퍼들(미도시) 내에 멀티플렉싱한다. 각 베퍼는 n 과 m 의 하나의 조합에 대한 FramePacketLength 계수들 C_n^m 을 포함한다.

[0310] 단계/스테이지(134)는 TrackNumberofOrderRegion + '1' 처리 경로를 초기화하고 디-인터리빙된 HOA 계수 베퍼들의 내용을 적절한 처리 경로에 전달한다. TrackRegion의 계수들은 HOA Track Header의 TrackRegionLastOrder 및 TrackRegionFirstOrder 필드에 의해 정의된다. 선택된 TrackRegions에 의해 커버되지 않는 HOA 차수들은, 포맷 변환 단계 또는 스테이지(136)와 잔여 계수 인터리빙 단계 또는 스테이지(135)를 포함한 표준 처리 경로에서 처리된다. 표준 처리 경로는 대역폭 감소 단계 또는 스테이지가 없는 TrackProcessing 경로에 대응한다.

[0311] TrackProcessing 경로에서, 포맷 변환 단계/스테이지(1331 내지 133N)는, TrackRegionSampleFormat에서 인코딩된 HOA 계수들을 디코더의 처리에 이용되는 데이터 포맷으로 변환한다. TrackRegionUseBandwidthReduction 데이터 필드에 따라, 대역폭이 감소되고 임계적으로 샘플링된 HOA 계수들이 Track의 전체 대역폭으로 복원되는 선택사항적인 대역폭 복원 단계 또는 스테이지(1321 내지 132N)가 후속된다. 복원 처리의 종류는 HOA Track Header의 TrackBandwidthReductionType 필드에서 정의된다.

[0312] 후속하는 인터리빙 단계 또는 스테이지(1311 내지 131N)에서, HOA 계수들의 디-인터리빙된 베퍼들의 내용은 하나의 시간 샘플의 HOA 계수들을 그룹화함으로써 인터리빙되고, 현재의 TrackRegion의 HOA 계수들은 이전의 TrackRegions의 HOA 계수들과 결합된다. HOA 계수들의 결과적 시퀀스는 Track의 처리에 맞게 적합화될 수 있다. 또한, 인터리빙 단계/스테이지는 대역폭 감소를 이용하는 TrackRegions 과 대역폭 감소를 이용하지 않는 TrackRegions 사이의 지연을 처리하며, 이 지연은 선택된 TrackBandwidthReductionType 처리에 의존한다. 예를 들어, MDCT 처리는 FramePacketSize 샘플들의 지연을 추가하므로, 대역폭 감소가 없는 처리 경로의 인터리빙 단계/스테이지는 그들의 출력을 1 패킷만큼 지연시킬 것이다.

[0313] MDCT를 통한 대역폭 감소

[0314] 인코딩

[0315] 도 14는 MDCT(modified discrete cosine transform) 처리를 이용한 대역폭 저감을 도시한다. FramePacketSize 샘플들의 TrackRegion의 각 HOA 계수는 베퍼(1411 내지 141M)를 통해 대응하는 MDCT 윈도우 추가 단계 또는 스테이지(1421 내지 142M)로 전달된다. 각 입력 베퍼는 n 과 m 의 하나의 조합의 시간적 연속 HOA 계수들 C_n^m 을 포함한다, 즉, 하나의 베퍼는 $(\text{buffer}) C_n^m = [C_n^m(0), C_n^m(1), \dots, C_n^m(\text{FramePacketSize} - 1)]$ 로서 정의된다.

[0316] 베퍼의 개수 M 은 앰비소닉 컴포넌트들의 개수와 동일하다(차수 N 의 풀 3D 음장에 대해 $(N + 1)^2$). 베퍼 처리는, 이전의 베퍼 내용을 현재의 베퍼 내용과 결합하여 대응하는 단계 또는 스테이지(1431 내지 143M)에서 MDCT 처리를 위한 새로운 내용으로 만듦으로써 후속 MDCT 처리를 위한 50% 중복을 수행하고, 후속 베퍼 내용의 처리를 위해 현재의 베퍼 내용을 저장한다. MDCT 처리는 각 Frame의 시작부에서 재시작하는데, 이것은 현재 Frame의 Track의 모든 계수들은 이전 Frame의 지식없이 디코딩될 수 있다는 것을 의미하며, 현재 Frame의 마지막 베퍼 내용에 후속하여, 제로의 추가 베퍼 내용이 처리된다. 따라서, MDCT 처리된 TrackRegions은 하나의 여분 TrackPacket을 생성한다. 윈도우 추가 단계/스테이지에서, 대응하는 베퍼 내용은, 각 TrackRegion에 대한 $w(t)$ HOATrack 헤더 필드 TrackRegionWindowType에서 정의된, 선택된 윈도우 함수 $w(t)$ 로 곱해진다.

[0317] 수정된 이산 코사인 변환은, J.P. Princen, A.B. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation", IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-34, no.5, pages 1153-1161, October 1986.에서 처음 언급되었다. MDCT는 FramePacketSize 부대역들의 임계적으로 샘플링된 필터 뱅크를 나타내는 것으로 간주될 수 있고, 이것은 50% 입력 베퍼 중복을 요구한다.

입력 베퍼는 부대역 크기의 2배의 길이를 가진다. MDCT는 FramePacketSize와 동일한 T 를 갖는 이하의 방정식에 의해 정의된다:

$$C'_n^m(k) = \sum_{t=0}^{2T-1} w(t) C_n^m(t) \cos \left[\frac{\pi}{T} \left(t + \frac{T+1}{2} \right) \left(k + \frac{1}{2} \right) \right] \text{ for } 0 \leq k < T$$

[0318]

[0319] 계수들 $C'_n^m(k)$ 은 MDCT 빈이라 불린다. MDCT 계산은 고속 푸리에 변환을 이용하여 구현될 수 있다. 후속하는 주파수 영역 컷-아웃 단계 또는 스테이지(1441 내지 144M)에서, 베퍼 길이를 TrackRegionLastBin - TrackRegionFirstBin+1로 감소하기 위하여 $k < \text{TrackRegionFirstBin}$ 이고 $k > \text{TrackRegionLastBin}$ 인 모든 MDCT 빈들 $C'_n^m(k)$ 을 제거함으로써 대역폭 감소가 수행되고, 여기서, TrackRegionFirstBin은 TrackRegion에 대한 하위 커-오프 주파수이고, TrackRegionLastBin은 상위 커-오프 주파수이다. MDCT 빈들의 무시는, TrackRegionLastBin 및 TrackRegionFirstBin 주파수들에 대응하는 커-오프 주파수를 갖는 대역통과 필터를 나타내는 것으로 간주될 수 있다. 따라서, 요구되는 MDCT 빈들만이 전송된다.

[0320] 디코딩

[0321] 도 15는 MDCT 처리를 이용한 대역폭 디코딩 또는 복원을 도시하며, 여기서 대역폭 제한된 TrackRegions의 HOA 계수들은 Track의 전체 대역폭으로 복원된다. 이 대역폭 복원은 시간적으로 디-인터리빙된 HOA 계수들의 베퍼

$$C_n^m(k)$$

내용을 병렬로 처리하고, 각 베피는 계수들을 $C_n^m(k)$ 의 TrackRegionLastBin - TrackRegionFirstBin + 1 MDCT 빈을 포함한다. 분실 주파수 영역 추가 단계(1541 내지 154M)은, 분실 MDCT 빈 $k < \text{TrackRegionFirstBin}$ 과 $k > \text{TrackRegionLastBin}$ 을 갖는 수신된 MDCT 빈들을 제로를 이용하여 보완함으로써 크기 FramePacketLength의 전체 MDCT 베피 내용을 복원한다.

$$C_n^m(t)$$

그 후, 시간 영역 HOA 계수들을 $C_n^m(t)$ 을 복원하기 위하여 대응하는 역 MDCT 단계 또는 스테이지(1531 내지 153M)에서 역 MDCT가 수행된다.

[0322]

역 MDCT는 합성 필터 뱅크로서 해석될 수 있고, 여기서 FramePacketLength MDCT 빈들은 2배의 FramePacketLength 시간 영역 계수들로 변환된다. 그러나, 시간 영역 샘플들의 완전한 복원은 인코더에서 사용

$$w(t)$$

된 윈도우 함수 $w(t)$ 와 현재의 베피 내용의 처음 절반과 이전 베피 내용의 두 번째 절반과의 중복-가산(overlap-add)과의 곱을 요구한다. 역 MDCT는 이하의 수학식에 의해 정의된다:

$$C_n^m(t) = \frac{w(t)}{2T} \sum_{k=0}^{T-1} C_n^m(k) \cos \left[\frac{\pi}{T} \left(t + \frac{T+1}{2} \right) \left(k + \frac{1}{2} \right) \right] \text{ for } 0 \leq t < T$$

[0323]

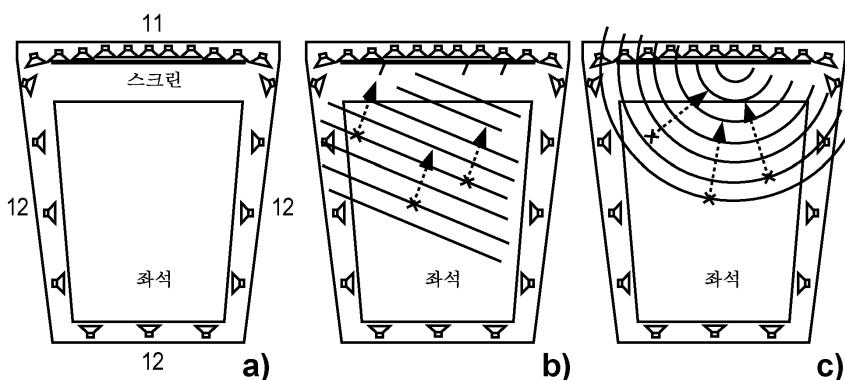
MDCT와 같이, 역 MDCT는 고속 역 푸리에 변환을 이용하여 구현될 수 있다. MDCT 윈도우 추가 단계 또는 스테이지(1521 내지 152M)는 복원된 시간 영역 계수들을 TrackRegionWindowType에 의해 정의된 윈도우 함수로 곱한다. FramePacketSize 시간 영역 계수들을 복원하기 위하여 후속 베피들(1511 내지 151M)은 현재의 TrackPacket 베피 내용의 처음 절반을 마지막 TrackPacket 베피 내용의 두 번째 절반에 추가한다. 현재 TrackPacket 베피 내용의 두 번째 절반은 후속 TrackPacket의 처리를 위해 저장되고, 이 중복-추가 처리는 양쪽 베피 내용의 상반되는 에일리어싱 컴포넌트들을 제거한다.

[0325]

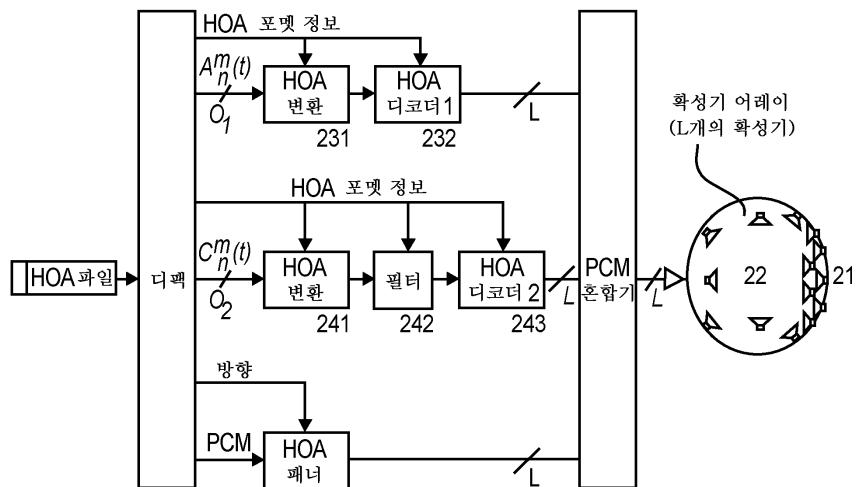
멀티-Frame HOA 파일들의 경우, 인코더는 새로운 Frame의 시작부에서의 중복-추가 프로시저를 위해 이전 프레임의 마지막 베피 내용을 이용하는 것이 금지된다. 따라서, Frame 경계 또는 새로운 Frame의 시작부에서, 중복-추가 베피 내용은 분실되고, Frame의 첫 번째 TrackPacket의 복원은 두 번째 TrackPacket에서 수행될 수 있고, 이로써 대역폭 감소없는 처리 경로에 비해 하나의 FramePacket의 지연과 하나의 여분의 TrackPacket의 디코딩이 도입된다. 이 지연은 도 13과 연계하여 설명된 인터리빙 단계/스테이지에 의해 처리된다.

도면

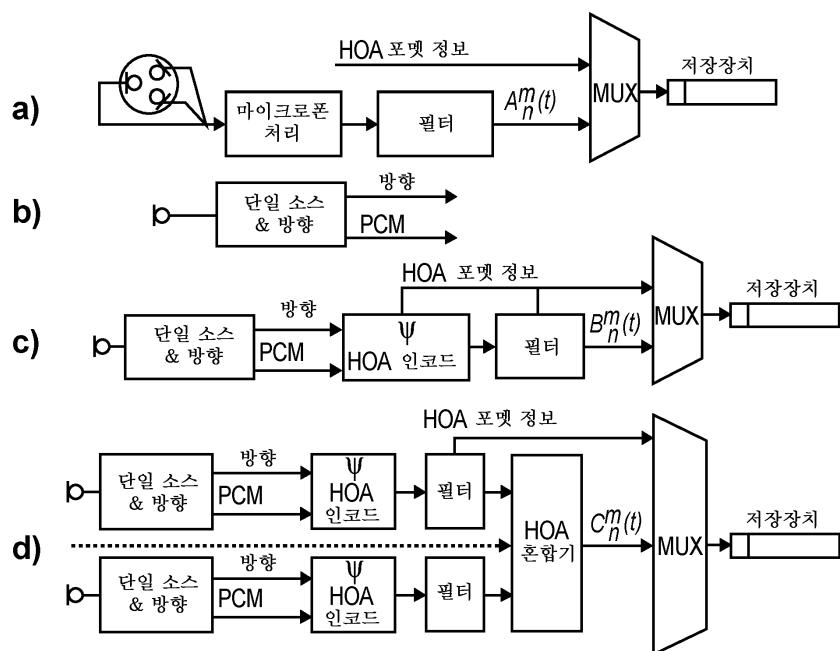
도면1



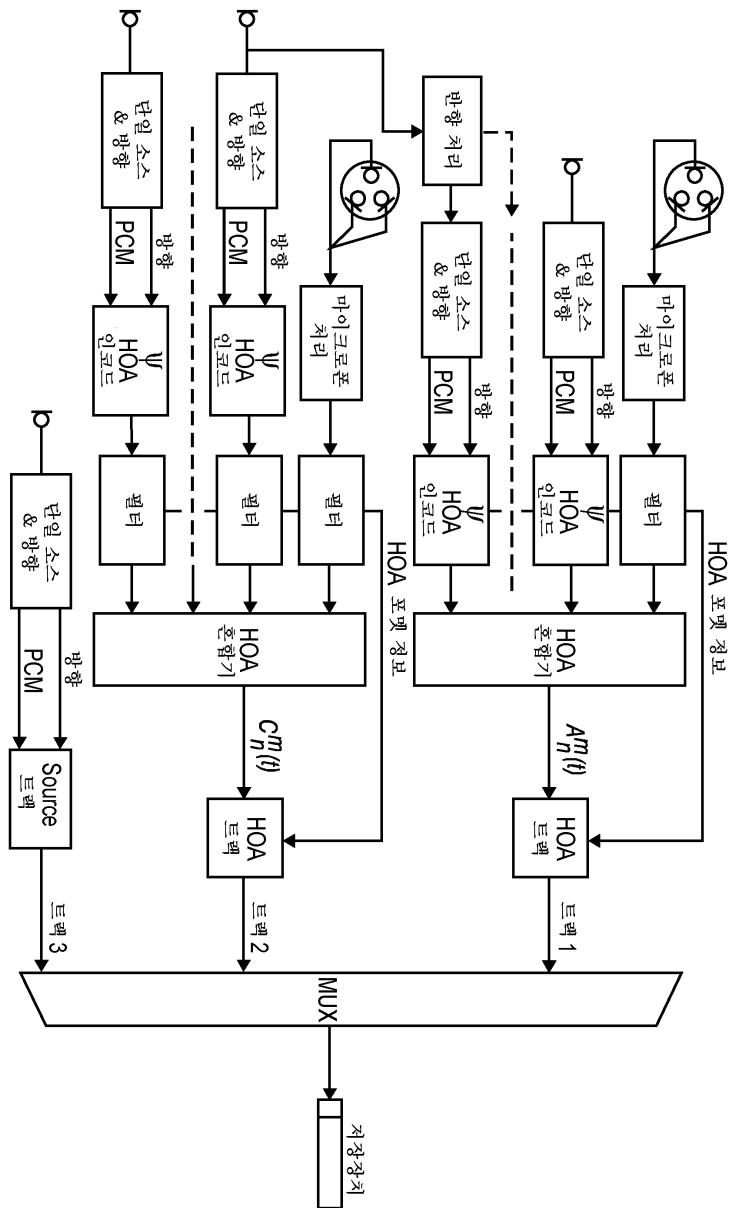
도면2



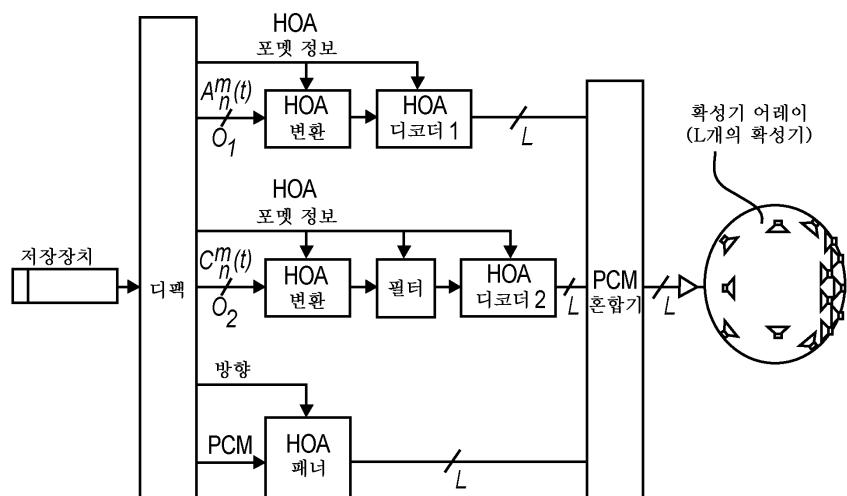
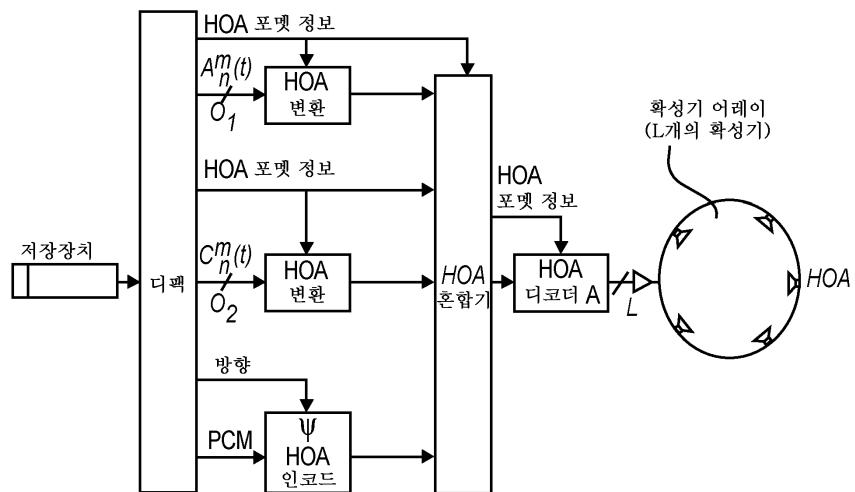
도면3



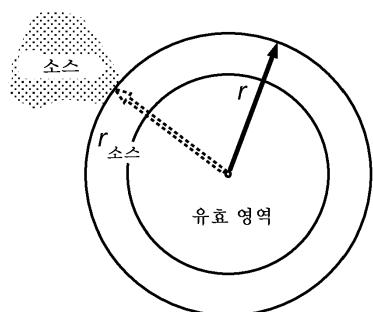
도면4



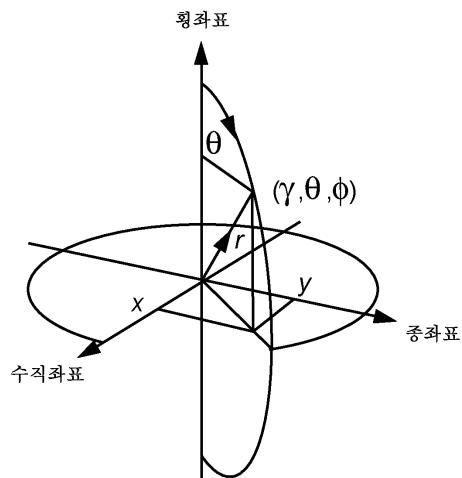
도면5



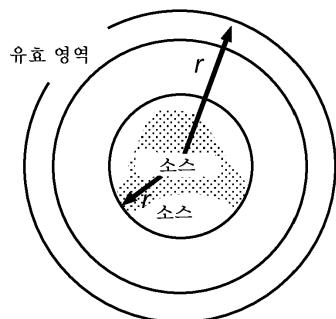
도면6



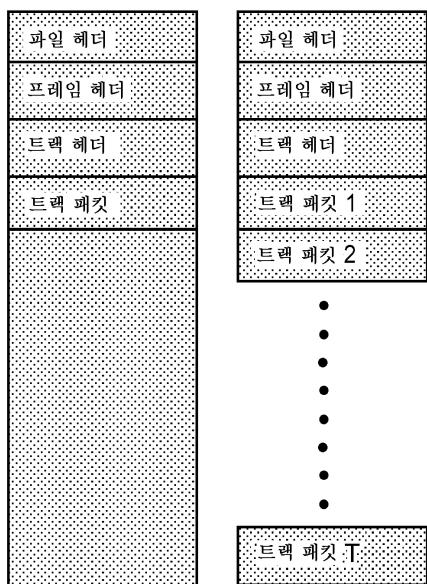
도면7



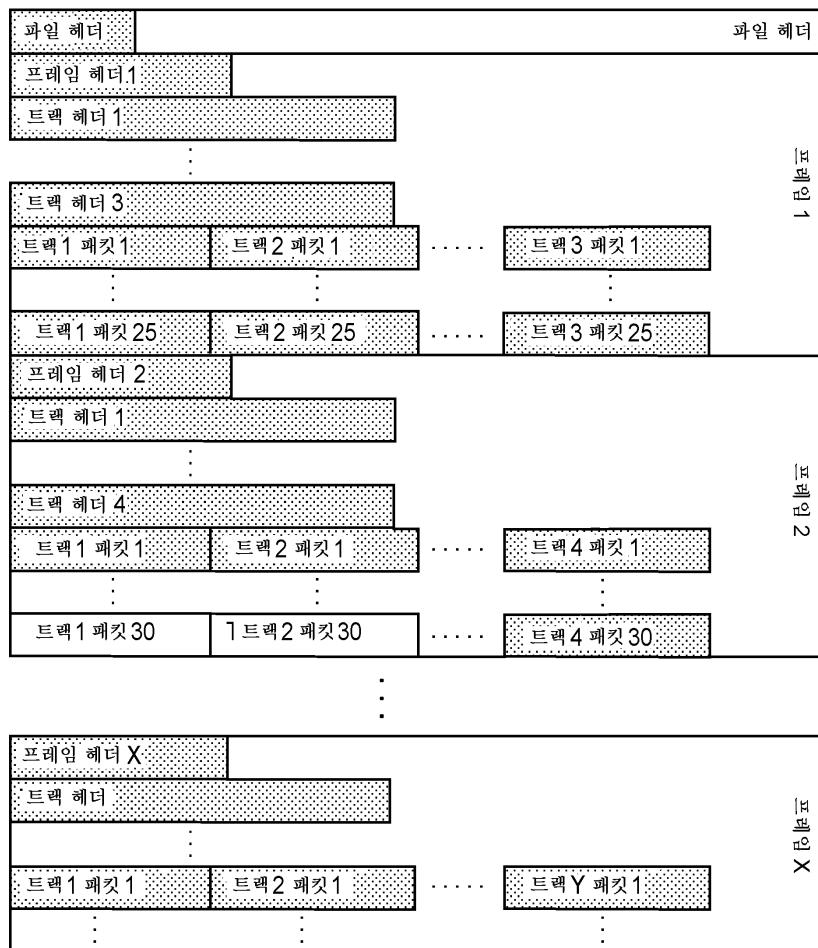
도면8



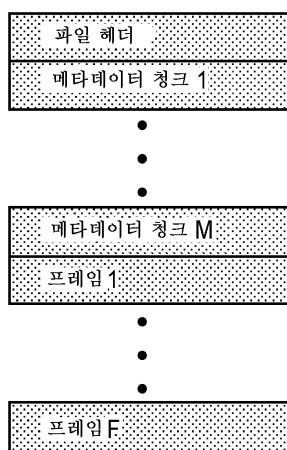
도면9



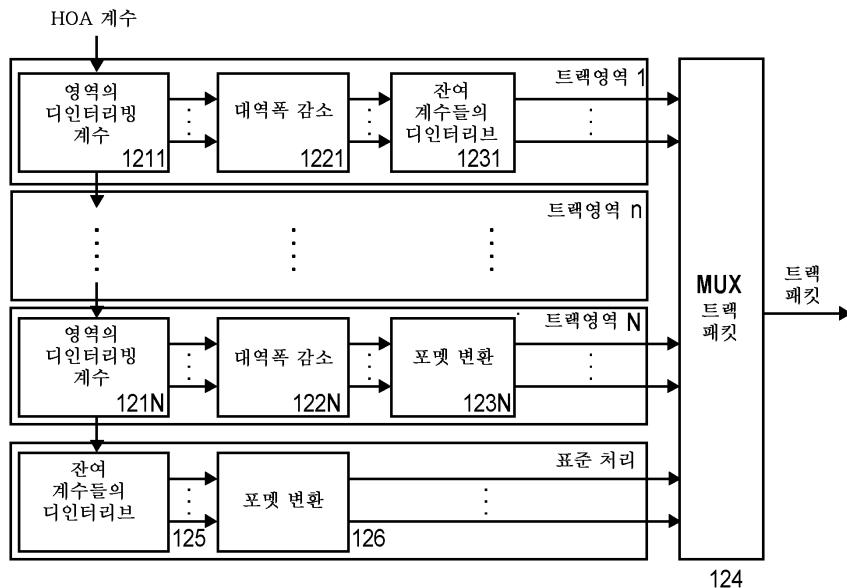
도면10



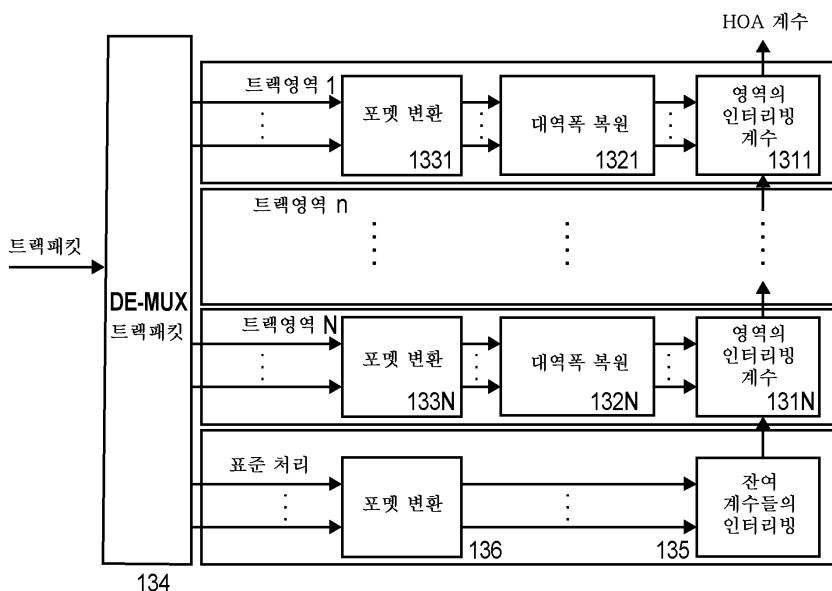
도면11



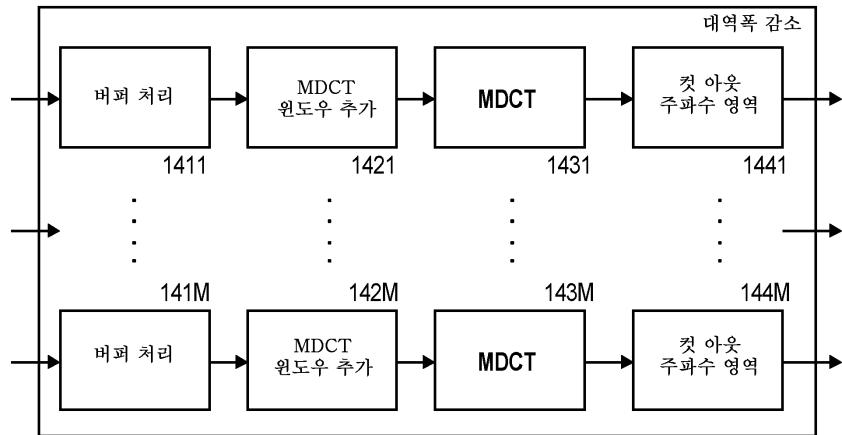
도면12



도면13



도면14



도면15

