



등록특허 10-2109159



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2020년05월12일  
(11) 등록번호 10-2109159  
(24) 등록일자 2020년05월04일

- (51) 국제특허분류(Int. Cl.)  
*G10L 19/008* (2014.01)
- (52) CPC특허분류  
*G10L 19/008* (2013.01)
- (21) 출원번호 10-2017-7028549
- (22) 출원일자(국제) 2016년03월08일  
심사청구일자 2017년10월11일
- (85) 번역문제출일자 2017년10월10일
- (65) 공개번호 10-2017-0130458
- (43) 공개일자 2017년11월28일
- (86) 국제출원번호 PCT/EP2016/054900
- (87) 국제공개번호 WO 2016/142375  
국제공개일자 2016년09월15일
- (30) 우선권주장  
15158234.3 2015년03월09일  
유럽특허청(EPO)(EP)  
15172492.9 2015년06월17일  
유럽특허청(EPO)(EP)
- (56) 선행기술조사문헌  
US20040049379 A1\*  
US20090112606 A1  
US20130077793 A1\*  
J.Breebaart, et al. MPEG spatial audio coding/MPEG surround: overview and current status. Audio Engineering Society Convention 119. 2005.10.10.
- \*는 심사관에 의하여 인용된 문헌

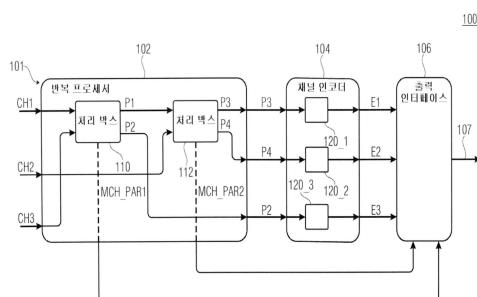
전체 청구항 수 : 총 24 항

심사관 : 이남숙

(54) 발명의 명칭 **다채널 신호를 인코딩 또는 디코딩하기 위한 장치 및 방법****(57) 요약**

실시예들은 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 장치를 제공한다. 이 장치는 반복 프로세서, 채널 인코더 및 출력 인터페이스를 포함한다. 반복 프로세서는 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 그리고 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 선택

(뒷면에 계속)

**대 표 도**

된 쌍에 대한 제 1 다채널 파라미터들을 도출하고 제 1 처리된 채널들을 도출하기 위해, 제 1 반복 단계에서 적어도 3개의 채널들의 각각의 쌍 사이의 채널 간 상관 값들을 계산하도록 구성된다. 또한, 반복 프로세서는 제 2 다채널 파라미터들 및 제 2 처리된 채널들을 도출하기 위해, 처리된 채널들 중 적어도 하나를 사용하여 제 2 반복 단계에서 계산, 선택 및 처리를 수행하도록 구성된다. 채널 인코더는 인코딩된 채널들을 획득하기 위해, 반복 프로세서에 의해 수행되는 반복 처리로부터 야기되는 채널들을 인코딩하도록 구성된다. 출력 인터페이스는 인코딩된 채널들과 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 생성하도록 구성된다.

## (72) 발명자

**레텔바치, 니콜라우스**

독일 뉘른베르크 90427, 슈페사르트스트리트 38

**슈베글러, 토비아스**

독일 뉘른베르크 90419, 이말리엔스트리트 35

**페그, 리차드**

독일 뉘른베르크 90419, 키르헨웨그 26

**힐퍼트, 요하네스**

독일 뉘른베르크 90411, 지에겔스테인스트라쎄 183

**뉴싱거, 마티아스**

독일 로어 91186, 베르그스트라베 10

## 명세서

### 청구범위

#### 청구항 1

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100)로서,

제 1 반복 단계에서 상기 적어도 3개의 채널들(CH1:CH3)의 각각의 쌍 사이의 채널 간 상관 값들을 계산하고, 상기 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하고, 그리고 선택된 쌍을 다채널 처리 연산(110, 112)을 사용하여 처리하여 상기 선택된 쌍에 대한 제 1 다채널 파라미터들(MCH\_PAR1)을 도출하고 처리된 채널들(P1, P2)의 제 1 쌍을 도출하기 위한 반복 프로세서(102) – 상기 반복 프로세서(102)는 제 2 다채널 파라미터들(MCH\_PAR2) 및 처리된 채널들(P3, P4)의 제 2 쌍을 도출하기 위해, 상기 적어도 3개의 채널들(CH1:CH3) 중 처리되지 않은 채널들 및 상기 처리된 채널들(P1, P1)을 사용하여 제 2 반복 단계에서 상기 계산, 상기 선택 및 상기 처리를 수행하도록 구성되고, 상기 반복 프로세서(102)는 상기 제 2 반복 단계에서 그리고 적용 가능하다면 임의의 추가 반복 단계들에서 상기 제 1 반복 단계의 선택된 쌍을 선택하지 않도록 구성됨 –;

인코딩된 채널들(E1:E3)을 획득하기 위해, 상기 반복 프로세서(102)에 의해 수행되는 반복 처리로부터 야기되는 채널들(P2:P4)을 인코딩하기 위한 채널 인코더 – 상기 반복 처리로부터 야기되어 상기 채널 인코더에 제공되는 채널들(P2:P4)의 수는 상기 반복 프로세서(102)에 입력되는 채널들(CH1:CH3)의 수와 같음 –; 및

상기 인코딩된 채널들(E1:E3)과 상기 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 생성하기 위한 출력 인터페이스(106)를 포함하며,

상기 제 1 다채널 파라미터들(MCH\_PAR1)은 상기 제 1 반복 단계에 대한 상기 선택된 쌍에서의 채널들의 제 1 식별을 포함하고,

상기 제 2 다채널 파라미터들(MCH\_PAR2)은 상기 제 2 반복 단계의 선택된 쌍에서의 채널들의 제 2 식별을 포함하는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

#### 청구항 2

제 1 항에 있어서,

상기 출력 인터페이스(106)는 직렬 비트스트림으로서 상기 인코딩된 다채널 신호(107)를 생성하도록, 그리고 이에 따라 상기 제 2 다채널 파라미터들(MCH\_PAR2)이 상기 제 1 다채널 파라미터들(MCH\_PAR1) 전에 상기 인코딩된 신호에 있도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

#### 청구항 3

제 1 항에 있어서,

상기 반복 프로세서(102)는 상기 선택된 쌍으로부터의 회전각 계산을 이용한 회전 처리 및 예측 처리를 포함하는 그룹 중 적어도 하나를 포함하는 스테레오 처리를 수행하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

#### 청구항 4

제 1 항에 있어서,

상기 반복 프로세서(102)는 복수의 대역들에 대한 단일 채널 간 상관 값이 얻어지게 상기 복수의 대역들을 포함하는 각각의 채널의 프레임을 사용하여 채널 간 상관을 계산하도록 구성되고,

상기 반복 프로세서(102)는 상기 복수의 대역들 각각에 대해 상기 제 1 또는 제 2 다채널 파라미터들(MCH\_PAR1,

MCH\_PAR2)이 얻어지도록 상기 복수의 대역들 각각에 대한 다채널 처리를 수행하도록 구성되는,  
적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 5

제 1 항에 있어서,

상기 반복 프로세서(102)는 제 1 프레임에 대해, 복수의 선택된 쌍 표시들을 도출하도록 구성되고,

상기 출력 인터페이스(106)는 상기 제 1 프레임 다음의 제 2 프레임에 대해, 상기 제 2 프레임이 상기 제 1 프레임과 동일한 복수의 선택된 쌍 표시들을 가짐을 나타내는 유지 표시자를 상기 다채널 신호(107)에 포함하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 6

제 1 항에 있어서,

상기 반복 프로세서(102)는 쌍의 레벨 차이가 임계치보다 작을 때만 상기 쌍을 선택하도록 구성되며,

상기 임계치는 40dB 또는 25dB 또는 12dB보다 더 작거나 6dB보다 더 작은,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 7

제 1 항에 있어서,

상기 반복 프로세서(102)는 정규화된 정수 상관 값들을 계산하도록 구성되며,

상기 반복 프로세서(102)는 상기 상관 값이 0.2보다 클 때, 쌍을 선택하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 8

제 1 항에 있어서,

상기 반복 프로세서(102)는 상기 다채널 처리에서 스테레오 파라미터들을 계산하도록 구성되며,

상기 반복 프로세서(102)는 스테레오 파라미터가 스테레오 파라미터 양자화기로 정의된, 0으로 양자화된 임계치 보다 더 높은 대역들에서만 스테레오 처리를 수행하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 9

제 1 항에 있어서,

상기 반복 프로세서(102)는 상기 다채널 처리에서 회전각들을 계산하도록 구성되며,

상기 반복 프로세서(102)는 회전각이 0으로 양자화된 디코더 측 임계치보다 더 높은 대역들에서만 회전 처리를 수행하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 10

제 1 항에 있어서,

상기 반복 프로세서(102)는 반복 종료 기준에 도달할 때까지 반복 단계들을 수행하도록 구성되며,

상기 반복 종료 기준은 반복 단계들의 최대 개수가 상기 다채널 신호(101)의 채널들(CH1:CH3)의 총 개수와 동일하거나 2개 더 많은 것이고, 또는

상기 반복 종료 기준은 상기 채널 간 상관 값들이 상기 임계치보다 더 큰 값을 갖지 않을 때인, 적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 11

제 1 항에 있어서,

상기 반복 프로세서(102)는 상기 처리된 채널들(P1, P2)이 중간 채널(P1) 및 측면 채널(P2)이 되도록 상기 제 1 반복 단계에서 상기 다채널 처리를 사용하여 상기 선택된 쌍을 처리하도록 구성되며,

상기 반복 프로세서(102)는 상기 제 2 다채널 파라미터들(MCH\_PAR2) 및 제 2 처리된 채널들(P3, P4)을 도출하기 위해, 상기 처리된 채널들(P1, P2) 중 적어도 하나로서 상기 처리된 채널들(P1, P2) 중 상기 중간 채널(P1)만을 사용하여 상기 제 2 반복 단계에서 상기 계산, 상기 선택 및 상기 처리를 수행하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 12

제 1 항에 있어서,

상기 채널 인코더는 상기 반복 처리로부터 야기된 채널들(P2:P4)을 인코딩하기 위한 채널 인코더들(120\_1:120\_3)을 포함하고,

상기 채널 인코더들은 더 많은 에너지를 갖는 채널을 인코딩하기 위해서보다 더 적은 에너지를 갖는 채널을 인코딩하기 위해 더 적은 비트들이 사용되게 상기 채널들(P2:P4)을 인코딩하도록 구성되는,

적어도 3개의 채널들(CH1:CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(100).

### 청구항 13

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200)로서,

디코딩된 채널들(D1:D3)을 얻기 위해, 상기 인코딩된 채널들(E1:E3)을 디코딩하기 위한 채널 디코더(202); 및 처리된 채널들(P1\*, P2\*)을 얻기 위해 상기 제 2 다채널 파라미터들(MCH\_PAR2)에 의해 식별된 상기 디코딩된 채널들(D1:D3)의 제 2 쌍을 사용하여 그리고 상기 제 2 다채널 파라미터들(MCH\_PAR2)을 사용하여 다채널 처리를 수행하고, 상기 제 1 다채널 파라미터들(MCH\_PAR1)에 의해 식별된 채널들(D1:D3, P1\*, P2\*)의 제 1 쌍을 사용하여 그리고 상기 제 1 다채널 파라미터들(MCH\_PAR1)을 사용하여 추가 다채널 처리를 수행하기 위한 다채널 프로세서(204)를 포함하며,

상기 채널들의 제 1 쌍은 적어도 하나의 처리된 채널(P1\*, P2\*)을 포함하고,

상기 다채널 처리로부터 야기되어 상기 다채널 프로세서(204)에 의해 출력되는 처리된 채널들의 수는 상기 다채널 프로세서(204)에 입력되는 디코딩된 채널들(D1:D3)의 수와 같고,

상기 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 각각 채널 쌍 식별을 포함하며,

상기 다채널 프로세서(204)는 미리 정의된 디코딩 규칙 또는 상기 인코딩된 다채널 신호에서 표시된 디코딩 규칙을 사용하여 채널 쌍 식별들을 디코딩하도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 14

제 13 항에 있어서,

상기 인코딩된 다채널 신호(107)는 제 1 프레임에 대해 상기 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 그리고 상기 제 1 프레임 다음의 제 2 프레임에 대해 유지 표시자를 포함하며,

상기 다채널 프로세서(204)는 상기 제 1 프레임에 사용된 것과 동일한 제 2 쌍 또는 동일한 제 1 쌍의 채널들에

대해 상기 제 2 프레임에서 상기 다채널 처리 또는 추가 다채널 처리를 수행하도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 15

제 13 항에 있어서,

상기 다채널 처리 및 상기 추가 다채널 처리는 스테레오 파라미터를 사용하는 스테레오 처리를 포함하며,

상기 디코딩된 채널들(D1:D3)의 개개의 스케일 팩터 대역들 또는 스케일 팩터 대역들의 그룹들에 대해, 제 1 스테레오 파라미터가 상기 제 1 다채널 파라미터(MCH\_PAR1)에 포함되고 제 2 스테레오 파라미터는 상기 제 2 다채널 파라미터(MCH\_PAR2)에 포함되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 16

제 13 항에 있어서,

상기 제 1 또는 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 어느 스케일 팩터 대역들이 다채널 처리되고 어떤 스케일 팩터 대역들이 다채널 처리되지 않는지를 나타내는 다채널 처리 마스크를 포함하고,

상기 다채널 프로세서(204)는 상기 다채널 처리 마스크에 의해 표시된 스케일 팩터 대역들에서 상기 다채널 처리를 수행하지 않도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 17

제 13 항에 있어서,

상기 디코딩 규칙은 허프만 디코딩 규칙이며,

상기 다채널 프로세서(204)는 상기 채널 쌍 식별들의 허프만 디코딩을 수행하도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 18

제 13 항에 있어서,

상기 인코딩된 다채널 신호(107)는 상기 다채널 처리가 허용되는 상기 디코딩된 채널들의 하위 그룹만을 표시하고 상기 다채널 처리가 허용되지 않는 적어도 하나의 디코딩된 채널을 표시하는 다채널 처리 허용 표시자를 포함하고,

상기 다채널 프로세서(204)는 상기 다채널 처리 허용 표시자에 의해 표시된 바와 같이 상기 다채널 처리가 허용되지 않는 적어도 하나의 디코딩된 채널에 대해 어떠한 다채널 처리도 수행하지 않도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

### 청구항 19

제 13 항에 있어서,

상기 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 스테레오 파라미터들을 포함하며,

상기 스테레오 파라미터들은 차동적으로 인코딩되며,

상기 다채널 프로세서(204)는 차동적으로 인코딩된 스테레오 파라미터들을 차동적으로 디코딩하기 위한 차동 디코더를 포함하는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

## 청구항 20

제 13 항에 있어서,

상기 인코딩된 다채널 신호(107)는 직렬 신호이며, 상기 제 2 다채널 파라미터들(MCH\_PAR2)은 상기 제 1 다채널 파라미터들(MCH\_PAR1) 전에 디코더(200)에서 수신되고,

상기 다채널 프로세서(204)는 상기 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)이 상기 디코더(200)에 의해 수신되는 순서로 상기 디코딩된 채널들(D1:D3)을 처리하도록 구성되는,

인코딩된 채널들(E1:E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(200).

## 청구항 21

적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법(300)으로서,

제 1 반복 단계에서 상기 적어도 3개의 채널들의 각각의 쌍 사이의 채널 간 상관 값을 계산하여, 상기 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하고, 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 상기 선택된 쌍에 대한 제 1 다채널 파라미터들을 도출하고 제 1 처리된 채널들을 도출하는 단계(302),

제 2 다채널 파라미터들 및 제 2 처리된 채널들을 도출하기 위해, 상기 적어도 3개의 채널들(CH1:CH3) 중 처리되지 않은 채널들 및 상기 처리된 채널들을 사용하여 제 2 반복 단계에서 상기 계산, 상기 선택 및 상기 처리를 수행하는 단계(304) – 상기 제 1 반복 단계의 상기 선택된 쌍은 상기 제 2 반복 단계에서 그리고 적용 가능하다면 임의의 추가 반복 단계들에서 선택되지 않음 –;

인코딩된 채널들을 획득하기 위해 반복 처리로부터 야기되는 채널들을 인코딩하는 단계(306) – 상기 반복 처리로부터 야기되는 채널들의 수는 상기 반복 처리가 수행되는 채널들의 수와 같음 –; 및

상기 인코딩된 채널들과 상기 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 생성하는 단계(308)를 포함하며,

상기 제 1 다채널 파라미터들(MCH\_PAR1)은 상기 제 1 반복 단계에 대한 상기 선택된 쌍에서의 채널들의 제 1 식별을 포함하고,

상기 제 2 다채널 파라미터들(MCH\_PAR2)은 상기 제 2 반복 단계의 선택된 쌍에서의 채널들의 제 2 식별을 포함하는,

적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법(300).

## 청구항 22

인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하는 방법(400)으로서,

디코딩된 채널들을 획득하기 위해, 상기 인코딩된 채널들을 디코딩하는 단계(402); 및

처리된 채널들을 얻기 위해 상기 제 2 다채널 파라미터들에 의해 식별된 상기 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 상기 제 2 다채널 파라미터들을 사용하여 다채널 처리를 수행하고, 상기 제 1 다채널 파라미터들에 의해 식별된 채널들의 제 1 쌍을 사용하여 그리고 상기 제 1 다채널 파라미터들을 사용하여 추가 다채널 처리를 수행하는 단계(404)를 포함하며,

상기 채널들의 제 1 쌍은 적어도 하나의 처리된 채널을 포함하고,

상기 다채널 처리로부터 야기되는 처리된 채널들의 수는 상기 다채널 처리가 수행되는 디코딩된 채널들의 수와

같으며,

상기 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 각각 채널 쌍 식별을 포함하고,

상기 채널 쌍 식별들은 미리 정의된 디코딩 규칙 또는 상기 인코딩된 다채널 신호에서 표시된 디코딩 규칙을 사용하여 디코딩되는,

인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하는 방법(400).

### 청구항 23

저장 매체에 저장된 컴퓨터 프로그램으로서,

컴퓨터 또는 프로세서 상에서 실행될 때, 제 21 항의 다채널 신호를 인코딩하는 방법, 또는 제 22 항의 인코딩된 다채널 신호를 디코딩하는 방법을 수행하기 위한,

저장 매체에 저장된 컴퓨터 프로그램.

### 청구항 24

제 1 항 또는 제 13 항의 장치로서,

다채널 처리는 조인트 스테레오 처리를 의미하고,

다채널 신호는 2개 보다 많은 채널들을 갖는,

장치.

### 청구항 25

삭제

### 청구항 26

삭제

### 청구항 27

삭제

## 발명의 설명

### 기술 분야

[0001]

본 발명은 오디오 코딩/디코딩에 관한 것으로, 특히 채널 간 신호 의존성을 활용하는 오디오 코딩에 관한 것이다.

### 배경 기술

[0002]

오디오 코딩은 오디오 신호들에서 중복성 및 부적절함의 활용을 다루는 압축 영역이다. MPEG USAC[ISO/IEC 23003-3:2012 - Information technology - MPEG audio technologies Part 3: Unified speech and audio coding]에서, 두 채널들의 조인트 스테레오 코딩은 복소 예측, MPS 2-1-2 또는 대역 제한 또는 전대역 잔여 신호들에 의한 통합 스테레오를 사용하여 수행된다. MPEG 서라운드[ISO/IEC 23003-1:2007 - Information technology - MPEG audio technologies Part 1: MPEG Surround]는 잔여 신호를 송신하거나 송신하지 않고 다채널 오디오의 조인트 코딩을 위해 OTT 및 TTT 박스들을 계층적으로 결합한다. MPEG-H 퀘드 채널 엘리먼트들은 MPS 2-1-2 스테레오 박스들을 계층적으로 적용한 다음 고정  $4 \times 4$  리믹싱 트리를 구축하는 복소 예측/MS 스테레오 박스들을 적용한다. AC4[ETSI TS 103 190 V1. 1 (2014-04) - Digital Audio Compression (AC-4) Standard]는 송신된 믹스 행렬 및 후속 조인트 스테레오 코딩 정보를 통해 송신된 채널들의 리믹싱을 가능하게 하는 새로운 3-, 4- 및 5-채널 엘리먼트들을 소개한다. 또한, 이전의 공보들은 개선된 다채널 오디오 코딩을 위해 카루넨 루베 변환(KLT: Karhunen-Loeve Transform)과 같은 직교 변환들을 사용하는 것을 제안한다[Yang, Dai and Ai, Hongmei and Kyriakakis, Chris and Kuo, C. -C. Jay, 2001: Adaptive Karhunen-Loeve Transform

for Enhanced Multichannel Audio Coding, <http://ict.usc.edu/pubs/Adaptive%20Karhunen-Loeve%20Transform%20for%20Enhanced%20Multichannel%20Audio%20Coding.pdf> ].

[0003] 3D 오디오 컨텍스트에서, 라우드스피커 채널들은 여러 높이의 레이어들로 분산되어, 수평 및 수직 채널 쌍들을 야기한다. USAC에 정의된 단 2개의 채널들의 조인트 코딩은 채널들 간의 공간 및 지각적 관계들을 고려하기에 충분하지 않다. MPEG 서라운드는 추가적인 전/후 처리 단계에서 적용되며, 잔여 신호들은 조인트 스테레오 코딩의 가능성 없이 개별적으로 송신되어, 예컨대 좌측 수직 잔여 신호와 우측 수직 잔여 신호 사이의 의존성을 활용한다. AC-4에는, 조인트 코딩 파라미터들의 효율적인 인코딩을 가능하게 하지만 새로운 몰입형 재생 시나리오들(7.1 + 4, 22.2)에 제안된 것처럼 더 많은 채널들을 갖춘 일반 스피커 설정들에는 실패하는 전용 N-채널 엘리먼트들이 도입된다. MPEG-H 쿼드 채널 엘리먼트는 또한 4개의 채널들로만 제한되며 임의의 채널들에 동적으로 적용될 수 있는 것이 아니라 사전 구성된 그리고 고정된 수의 채널들에만 동적으로 적용될 수 있다.

### 발명의 내용

[0004] 본 발명의 목적은 개선된 인코딩/디코딩 개념을 제공하는 것이다.

[0005] 이 목적은 청구항 제 1 항에 따른 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 장치, 청구항 제 12 항에 따른 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 장치, 청구항 제 21 항에 따른 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법, 제 22 항에 따른 인코딩된 채널들과 적어도 제 1 및 제 2 채널 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 복호화하기 위한 방법, 또는 청구항 제 23 항에 따른 컴퓨터 프로그램에 의해 달성된다.

[0006] 실시예들은 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 장치를 제공한다. 이 장치는 반복 프로세서, 채널 인코더 및 출력 인터페이스를 포함한다. 반복 프로세서는 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 그리고 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 선택된 쌍에 대한 제 1 다채널 파라미터들을 도출하고 제 1 처리된 채널들을 도출하기 위해, 제 1 반복 단계에서 적어도 3개의 채널들의 각각의 쌍 사이의 채널 간 상관 값을 계산하도록 구성된다. 또한, 반복 프로세서는 제 2 다채널 파라미터들 및 제 2 처리된 채널들을 도출하기 위해, 처리된 채널들 중 적어도 하나를 사용하여 제 2 반복 단계에서 계산, 선택 및 처리를 수행하도록 구성된다. 채널 인코더는 인코딩된 채널들을 획득하기 위해, 반복 프로세서에 의해 수행되는 반복 처리로부터 야기되는 채널들을 인코딩하도록 구성된다. 출력 인터페이스는 인코딩된 채널들과 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 생성하도록 구성된다.

[0007] 추가 실시예들은 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 장치를 제공한다. 이 장치는 채널 디코더 및 다채널 프로세서를 포함한다. 채널 디코더는 디코딩된 채널들을 얻기 위해, 인코딩된 채널들을 디코딩하도록 구성된다. 다채널 프로세서는 처리된 채널들을 얻기 위해 제 2 다채널 파라미터들에 의해 식별된 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 제 2 다채널 파라미터들을 사용하여 다채널 처리를 수행하고, 제 1 다채널 파라미터들에 의해 식별된 채널들의 제 1 쌍을 사용하여 그리고 제 1 다채널 파라미터들을 사용하여 추가 다채널 처리를 수행하도록 구성되며, 채널들의 제 1 쌍은 적어도 하나의 처리된 채널을 포함한다.

[0008] 고정된 신호 경로(예를 들어, 스테레오 코딩 트리)를 사용하는 일반적인 다채널 인코딩 개념들과는 대조적으로, 본 발명의 실시예들은 다채널 입력 신호의 적어도 3개의 입력 채널들의 특성에 적응된 동적 신호 경로를 사용한다. 상세하게는, 반복 프로세서(102)는 제 1 반복 단계에서 가장 높은 값 또는 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 적어도 3개의 채널들(CH1 내지 CH3)의 각각의 쌍 사이의 채널 간 상관 값을 기초하여 제 1 반복 단계에서, 그리고 제 2 반복 단계에서 가장 높은 값 또는 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 적어도 3개의 채널들의 각각의 쌍 및 대응하는 이전 처리된 채널들 간의 채널 간 상관 값을 기초하여 제 2 반복 단계에서, 신호 경로(예를 들어, 스테레오 트리)를 구축하도록 적응될 수 있다.

[0009] 추가 실시예들은 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법을 제공한다. 이 방법은,

[0010] 제 1 반복 단계에서 적어도 3개의 채널들의 각각의 쌍 사이의 채널 간 상관 값을 계산하여, 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하고, 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 선택된 쌍에 대한 제 1 다채널 파라미터들을 도출하고 제 1 처리된 채널들을 도출하는 단계;

[0011] 제 2 다채널 파라미터들 및 제 2 처리된 채널들을 도출하기 위해, 처리된 채널들 중 적어도 하나를 사용하여 제 2 반복 단계에서 계산, 선택 및 처리를 수행하는 단계;

- [0012] 인코딩된 채널들을 획득하기 위해, 반복 프로세서에 의해 수행되는 반복 처리로부터 야기되는 채널들을 인코딩하는 단계; 및
- [0013] 인코딩된 채널들과 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 생성하는 단계를 포함한다.
- [0014] 추가 실시예들은 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 방법을 제공한다. 이 방법은,
- [0015] 디코딩된 채널들을 획득하기 위해, 인코딩된 채널들을 디코딩하는 단계; 및
- [0016] 처리된 채널들을 얻기 위해 제 2 다채널 파라미터들에 의해 식별된 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 제 2 다채널 파라미터들을 사용하여 다채널 처리를 수행하고, 제 1 다채널 파라미터들에 의해 식별된 채널들의 제 1 쌍을 사용하여 그리고 제 1 다채널 파라미터들을 사용하여 추가 다채널 처리를 수행하는 단계를 포함하며, 채널들의 제 1 쌍은 적어도 하나의 처리된 채널을 포함한다.

### 도면의 간단한 설명

- [0017] 첨부 도면들을 참조로 여기서 본 발명의 실시예들이 설명된다.
- 도 1은 일 실시예에 따라, 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 장치의 개략적인 블록도를 도시한다.
- 도 2는 일 실시예에 따라, 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 장치의 개략적인 블록도를 도시한다.
- 도 3은 일 실시예에 따른 스테레오 박스의 개략적인 블록도를 도시한다.
- 도 4는 일 실시예에 따라, 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 장치의 개략적인 블록도를 도시한다.
- 도 5는 일 실시예에 따라, 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법의 흐름도를 도시한다.
- 도 6은 본 발명의 일 실시예에 따라, 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 방법의 흐름도를 도시한다.

### 발명을 실시하기 위한 구체적인 내용

- [0018] 동일한 또는 대등한 엘리먼트들 또는 동일한 또는 대등한 기능을 갖는 엘리먼트들은 다음 설명에서 동일한 또는 대등한 참조 번호들로 표시된다.
- [0019] 다음 설명에서는, 본 발명의 실시예들의 보다 철저한 설명을 제공하도록 복수의 세부사항들이 제시된다. 그러나 본 발명의 실시예들은 이러한 특정 세부사항들 없이 실시될 수도 있음이 해당 기술분야에서 통상의 지식을 가진 자들에게 명백할 것이다. 다른 경우들에는, 본 발명의 실시예들을 모호하게 하는 것을 피하기 위해, 잘 알려진 구조들 및 디바이스들은 상세히보다는 블록도 형태로 도시된다. 추가로, 구체적으로 달리 언급되지 않는 한, 이하 설명되는 서로 다른 실시예들의 특징들이 서로 결합될 수도 있다.
- [0020] 도 1은 적어도 3개의 채널들(CH1 내지 CH3)을 갖는 다채널 신호(101)를 인코딩하기 위한 장치(인코더)(100)의 개략적인 블록도를 도시한다. 이 장치(100)는 반복 프로세서(102), 채널 인코더(104) 및 출력 인터페이스(106)를 포함한다.
- [0021] 반복 프로세서(102)는 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 그리고 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 선택된 쌍에 대한 제 1 다채널 파라미터들(MCH\_PAR1)을 도출하고 제 1 처리된 채널들(P1, P2)을 도출하기 위해, 제 1 반복 단계에서 적어도 3개의 채널들(CH1 내지 CH3)의 각각의 쌍 사이의 채널 간 상관 값을 계산하도록 구성된다. 또한, 반복 프로세서(102)는 제 2 다채널 파라미터들(MCH\_PAR2) 및 제 2 처리된 채널들(P3, P4)을 도출하기 위해, 처리된 채널들(P1 또는 P2) 중 적어도 하나를 사용하여 제 2 반복 단계에서 계산, 선택 및 처리를 수행하도록 구성된다.
- [0022] 예를 들어, 도 1에 나타낸 바와 같이, 반복 프로세서(102)는 제 1 반복 단계에서, 적어도 3개의 채널들(CH1 내지 CH3)의 제 1 쌍 사이의 채널 간 상관 값을 계산할 수 있으며, 제 1 쌍은 제 1 채널(CH1)과 제 2 채널(CH2),

적어도 3개의 채널들(CH1 내지 CH3)의 제 2 쌍 사이의 채널 간 상관 값으로 구성되고, 제 2 쌍은 제 2 채널(CH2)과 제 3 채널(CH3), 그리고 적어도 3개의 채널들(CH1 내지 CH3)의 제 3 쌍 사이의 채널 간 상관 값으로 구성되며, 제 3 쌍은 제 1 채널(CH1)과 제 3 채널(CH3)로 구성된다.

[0023] 도 1에서는, 제 1 반복 단계에서, 제 1 채널(CH1)과 제 3 채널(CH3)로 구성된 제 3 쌍은 가장 높은 채널 간 상관 값을 포함하여, 반복 프로세서(102)가 제 1 반복 단계에서 가장 높은 채널 간 상관 값을 갖는 제 3 쌍을 선택하고 선택된 쌍, 즉 제 3 쌍을 다채널 처리 연산을 사용해 처리하여, 선택된 쌍에 대한 제 1 다채널 파라미터들(MCH\_PAR1)을 도출하고 제 1 처리된 채널들(P1, P2)을 도출한다고 가정된다.

[0024] 또한, 반복 프로세서(102)는 제 2 반복 단계에서 가장 높은 채널 간 상관 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해, 제 2 반복 단계에서 적어도 3개의 채널들(CH1 내지 CH3)의 각각의 쌍과 처리된 채널들(P1, P2) 사이의 채널 간 상관 값을 계산하도록 구성된다. 이로써, 반복 프로세서(102)는 제 2 반복 단계에서(또는 임의의 추가 반복 단계에서) 제 1 반복 단계의 선택된 쌍을 선택하지 않도록 구성될 수 있다.

[0025] 도 1에 도시된 예를 참조하면, 반복 프로세서(102)는 제 1 채널(CH1)과 제 1 처리된 채널(P1)로 구성된 채널들의 제 4 쌍 사이의 채널 상관 값, 제 1 채널(CH1)과 제 2 처리된 채널(P2)로 구성된 제 5 쌍 사이의 채널 상관 값, 제 2 채널(CH2)과 제 2 처리된 채널(P2)로 구성된 제 6 쌍 사이의 채널 상관 값, 제 2 채널(CH2)과 제 2 처리된 채널(P2)로 구성된 제 7 쌍 사이의 채널 상관 값, 제 3 채널(CH3)과 제 1 처리된 채널(P1)로 구성된 제 8 쌍 사이의 채널 상관 값, 제 3 채널(CH3)과 제 2 처리된 채널(P2)로 구성된 제 9 쌍 사이의 채널 상관 값, 및 제 1 처리된 채널(P1)과 제 2 처리된 채널(P2)로 구성된 제 10 쌍 사이의 채널 간 상관 값을 추가로 계산할 수 있다.

[0026] 도 1에서는, 제 2 반복 단계에서, 제 2 채널(CH2)과 제 1 처리된 채널(P1)로 구성된 제 6 쌍은 가장 높은 채널 간 상관 값을 포함하여, 반복 프로세서(102)가 제 2 반복 단계에서 제 6 쌍을 선택하고 선택된 쌍, 즉 제 6 쌍을 다채널 처리 연산을 사용해 처리하여, 선택된 쌍에 대한 제 2 다채널 파라미터들(MCH\_PAR2)을 도출하고 제 2 처리된 채널들(P3, P4)을 도출한다고 가정된다.

[0027] 반복 프로세서(102)는 쌍의 레벨 차이가 임계치보다 작을 때만 그 쌍을 선택하도록 구성될 수 있는데, 임계치는 40dB, 25dB, 12dB보다 더 작거나 6dB보다 더 작다. 이로써, 25 또는 40dB의 임계치들은 3도 또는 0.5도의 회전 각들에 해당한다.

[0028] 반복 프로세서(102)는 정규화된 정수 상관 값을 계산하도록 구성될 수 있으며, 여기서 반복 프로세서(102)는 정수 상관 값이 예를 들어, 0.2 또는 바람직하게는 0.3보다 클 때, 쌍을 선택하도록 구성될 수 있다.

[0029] 또한, 반복 프로세서(102)는 채널 인코더(104)에 다채널 처리로부터 야기되는 채널들을 제공할 수 있다. 예를 들어, 도 1을 참조하면, 반복 프로세서(102)는 제 2 반복 단계에서 수행된 다채널 처리로부터 야기된 제 3 처리된 채널(P3) 및 제 4 처리된 채널(P4) 그리고 제 1 반복 단계에서 수행된 다채널 처리로부터 야기된 제 2 처리된 채널(P2)을 채널 인코더(104)에 제공할 수 있다. 이로써, 반복 프로세서(102)는 후속 반복 단계에서(추가) 처리되지 않은 그러한 처리된 채널들만을 채널 인코더(104)에 제공할 수 있다. 도 1에 도시된 바와 같이, 제 1 처리된 채널(P1)은 제 2 반복 단계에서 추가 처리되기 때문에 이는 채널 인코더(104)에 제공되지 않는다.

[0030] 채널 인코더(104)는 반복 프로세서(102)에 의해 수행된 반복 처리(또는 다채널 처리)로부터 야기된 채널들(P2 내지 P4)을 인코딩하여 인코딩된 채널들(E1 내지 E3)을 획득하도록 구성될 수 있다.

[0031] 예를 들어, 채널 인코더(104)는 반복 처리(또는 다채널 처리)로부터 야기된 채널들(P2 내지 P4)을 인코딩하기 위해 모노 인코더들(또는 모노 박스들 또는 모노 터틀들)(120\_1 내지 120\_3)을 사용하도록 구성될 수 있다. 모노 박스들은 더 많은 에너지(또는 더 높은 진폭)를 갖는 채널을 인코딩하기 위해서보다 더 적은 에너지(또는 더 작은 진폭)를 갖는 채널을 인코딩하기 위해 더 적은 비트들이 요구되게 채널들을 인코딩하도록 구성될 수 있다. 모노 박스들(120\_1 내지 120\_3)은 예를 들어 변환 기반 오디오 인코더들일 수 있다. 또한, 채널 인코더(104)는 반복 처리(또는 다채널 처리)로부터 야기된 채널들(P2 내지 P4)을 인코딩하기 위해 스테레오 인코더들(예를 들어, 파라메트릭 스테레오 인코더들 또는 손실 있는 스테레오 인코더들)을 사용하도록 구성될 수 있다.

[0032] 출력 인터페이스(106)는 인코딩된 채널들(E1 내지 E3)과 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 생성하도록 구성될 수 있다.

[0033] 예를 들어, 출력 인터페이스(106)는 직렬 신호 또는 직렬 비트스트림으로서 인코딩된 다채널 신호(107)를 생성하도록, 그리고 이에 따라 제 2 다채널 파라미터들(MCH\_PAR2)이 제 1 다채널 파라미터들(MCH\_PAR1) 전에 인코딩

된 신호(107)에 있도록 구성될 수 있다. 따라서 도 4와 관련하여 이후에 실시예가 설명될 디코더는 제 1 다채널 파라미터들(MCH-PAR1) 전에 제 2 다채널 파라미터들(MCH\_PAR2)을 수신할 것이다.

[0034] 도 1에서, 반복 프로세서(102)는 예시적으로 2개의 다채널 처리 연산들인 제 1 반복 단계에서의 다채널 처리 연산 및 제 2 반복 단계에서의 다채널 처리 연산을 수행한다. 당연히, 반복 프로세서(102)는 또한 후속 반복 단계들에서 추가 다채널 처리 연산들을 수행할 수 있다. 이로써, 반복 프로세서(102)는 반복 종료 기준에 도달할 때 까지 반복 단계들을 수행하도록 구성될 수 있다. 반복 종료 기준은 반복 단계들의 최대 개수가 다채널 신호(101)의 총 채널 수와 동일하거나 2개 더 많은 것일 수 있고, 또는 여기서 반복 종료 기준은 채널 간 상관 값들이 임계치보다 더 큰 값을 갖지 않을 때, 임계치가 바람직하게는 0.2보다 더 크거나 임계치가 바람직하게는 0.3인 것이다. 추가 실시예들에서, 반복 종료 기준은 반복 단계들의 최대 개수가 다채널 신호(101)의 총 채널 수와 동일하거나 더 많은 것일 수 있고, 또는 여기서 반복 종료 기준은 채널 간 상관 값들이 임계치보다 더 큰 값을 갖지 않을 때, 임계치가 바람직하게는 0.2보다 더 크거나 임계치가 바람직하게는 0.3인 것이다.

[0035] 설명의 목적으로, 제 1 반복 단계 및 제 2 반복 단계에서 반복 프로세서(102)에 의해 수행되는 다채널 처리 연산들은 도 1에 처리 박스들(110, 112)로 예시적으로 예시된다. 처리 박스들(110, 112)은 하드웨어 또는 소프트웨어로 구현될 수 있다. 처리 박스들(110, 112)은 예를 들어, 스테레오 박스들일 수 있다.

[0036] 이로써, 채널 간 신호 의존성을 공지된 조인트 스테레오 코딩 툴들을 계층적으로 적용함으로써 활용될 수 있다. 이전 MPEG 접근 방식들과는 달리, 처리될 신호 쌍들은 고정 신호 경로(예를 들어, 스테레오 코딩 트리)에 의해 미리 결정되는 것이 아니라, 입력 신호 특성들에 적응하도록 동적으로 변경될 수 있다. 실제 스테레오 박스의 입력들은 (1) 채널들(CH1 내지 CH3)과 같은 처리되지 않은 채널들, (2) 처리된 신호들(P1 내지 P4)과 같은 선행하는 스테레오 박스의 출력들, 또는 (3) 처리되지 않은 채널과 선행하는 스테레오 박스의 출력의 결합일 수 있다.

[0037] 스테레오 박스(110, 112) 내부에서의 처리는 (USAC의 복소 예측 박스와 같이) 예측 기반 또는 KLT/PCA 기반일 수 있다(입력 채널들이 인코더에서 (예컨대, 2x2 회전 행렬을 통해) 회전되어 에너지 압축을 최대화, 즉 신호 에너지를 하나의 채널로 집중시키고, 디코더에서는 회전된 신호들이 원래의 입력 신호 방향들로 재변환될 것이다).

[0038] 인코더(100)의 가능한 구현에서, (1) 인코더는 모든 채널 쌍 사이의 채널 간 상관을 계산하고 입력 신호들 중 하나의 적절한 신호 쌍을 선택하여 선택된 채널들에 스테레오 툴을 적용하고; (2) 인코더는 모든 채널들(처리되지 않은 채널들뿐만 아니라 처리된 중간 출력 채널들) 사이의 채널 간 상관을 재계산하고 입력 신호들 중 하나의 적절한 신호 쌍을 선택하여 선택된 채널들에 스테레오 툴을 적용하고; 그리고 (3) 인코더는 모든 채널 간 상관이 임계치 이하가 될 때까지 또는 최대 변환 횟수가 적용된다면 단계(2)를 반복한다.

[0039] 이미 언급한 바와 같이, 인코더(100) 또는 보다 정확하게 반복 프로세서(102)에 의해 처리될 신호 쌍들은 고정 신호 경로(예를 들어, 스테레오 코딩 트리)에 의해 미리 결정되는 것이 아니라, 입력 신호 특성들에 적응하도록 동적으로 변경될 수 있다. 이로써, 인코더(100)(또는 반복 프로세서(102))는 다채널(입력) 신호(101)의 적어도 3개의 채널들(CH1 내지 CH3)에 따라 스테레오 트리를 구성하도록 구성될 수 있다. 즉, 인코더(100)(또는 반복 프로세서(102))는 채널 간 상관에 기초하여(예를 들어, 제 1 반복 단계에서 가장 높은 값 또는 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 제 1 반복 단계에서 적어도 3개의 채널들(CH1 내지 CH3)의 각각의 쌍 사이의 채널 간 상관 값들을 계산함으로써, 그리고 제 2 반복 단계에서 가장 높은 값 또는 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 제 2 반복 단계에서 적어도 3개의 채널들의 각각의 쌍과 이전 처리된 채널들 사이의 채널 간 상관 값들을 계산함으로써) 스테레오 트리를 구축하도록 구성될 수 있다. 한 단계 접근 방식에 따라, 가능하게는 이전 반복들에서 처리된 모든 채널들의 상관들을 포함하는, 가능하게는 각각의 반복에 대해 상관 행렬이 계산될 수 있다.

[0040] 위에서 나타낸 바와 같이, 반복 프로세서(102)는 제 1 반복 단계에서 선택된 쌍에 대한 제 1 다채널 파라미터들(MCH\_PAR1)을 도출하고 제 2 반복 단계에서 선택된 쌍에 대한 제 2 다채널 파라미터들(MCH\_PAR2)을 도출하도록 구성될 수 있다. 제 1 다채널 파라미터들(MCH\_PAR1)은 제 1 반복 단계에서 선택된 채널들의 쌍을 식별(또는 시그널링)하는 제 1 채널 쌍 식별(또는 인덱스)을 포함할 수 있으며, 여기서 제 2 다채널 파라미터들(MCH\_PAR2)은 제 2 반복 단계에서 선택된 채널들의 쌍을 식별(또는 시그널링)하는 제 2 채널 쌍 식별(또는 인덱스)을 포함할 수 있다.

[0041] 다음에는, 입력 신호들의 효율적인 인덱싱이 설명된다. 예를 들어, 채널 쌍들은 총 채널 수에 따라 각각의 쌍에

대한 고유 인덱스를 사용하여 효율적으로 시그널링될 수 있다. 예를 들어, 6개의 채널들에 대한 쌍들의 인덱싱은 다음 표에 도시된 바와 같다:

[0042] [표 1]

	0	1	2	3	4	5
0	0	1	2	3	4	
1		5	6	7	8	
2			9	10	11	
3				12	13	
4					14	
5						

[0043]

예를 들어, 상기 표에서, 인덱스(5)는 제 1 채널 및 제 2 채널로 구성된 쌍을 시그널링할 수 있다. 마찬가지로, 인덱스(6)는 제 1 채널 및 제 3 채널로 구성된 쌍을 시그널링할 수 있다.

[0045]

n개의 채널들에 대해 가능한 채널 쌍 인덱스들의 총 수는 다음과 같이 계산될 수 있다:

[0046]

$$\text{numPairs} = \text{numChannels} * (\text{numChannels}-1)/2$$

[0047]

따라서 하나의 채널 쌍을 시그널링하기 위해 필요한 비트들의 수는 다음과 같다:

[0048]

$$\text{numBits} = \text{floor}(\log_2(\text{numPairs}-1))+1$$

[0049]

또한, 인코더(100)는 채널 마스크를 사용할 수 있다. 다채널 툴의 구성은 툴들이 활성화된 채널들을 나타내는 채널 마스크를 포함할 수 있다. 따라서 LFE(LFE = 저주파 효과/확장 채널들)가 채널 쌍 인덱싱에서 제거될 수 있어, 보다 효율적인 인코딩을 가능하게 한다. 예컨대, 11. 1 설정의 경우, 이는 채널 쌍 인덱스들의 수를  $12*11/2 = 66$ 에서  $11*10/2 = 55$ 로 줄여, 7 비트 대신 6 비트로의 시그널링을 가능하게 한다. 이 메커니즘은 모노 객체들(예컨대, 다수의 언어 트랙들)인 것으로 의도된 채널들을 제외하는 데에도 또한 사용될 수 있다. 채널 마스크(channelMask)의 디코딩에서, 디코더 채널들에 대한 채널 쌍 인덱스들의 재맵핑을 가능하게 하도록 채널 맵(channelMap)이 생성될 수 있다.

[0050]

더욱이, 반복 프로세서(102)는 제 1 프레임에 대해, 복수의 선택된 쌍 표시들을 도출하도록 구성될 수 있으며, 여기서 출력 인터페이스(106)는 제 1 프레임 다음의 제 2 프레임에 대해, 제 2 프레임이 제 1 프레임과 동일한 복수의 선택된 쌍 표시들을 가짐을 나타내는 유지 표시자를 다채널 신호(107)에 포함하도록 구성될 수 있다.

[0051]

유지 표시자 또는 유지 트리 플래그는 어떠한 새로운 트리도 송신되지 않지만 마지막 스테레오 트리가 사용될 것임을 시그널링하는 데 사용될 수 있다. 이것은 채널 상관 특성들이 더 오랜 시간 동안 고정된 상태라면 동일한 스테레오 트리 구성의 다중 송신을 피하는 데 사용될 수 있다.

[0052]

도 2는 스테레오 박스(110, 112)의 개략적인 블록도를 도시한다. 스테레오 박스(110, 112)는 제 1 입력 신호(I1) 및 제 2 입력 신호(I2)에 대한 입력들, 그리고 제 1 출력 신호(O1) 및 제 2 출력 신호(O2)에 대한 출력들을 포함한다. 도 2에 나타낸 바와 같이, 입력 신호들(I1, I2)로부터의 출력 신호들(O1, O2)의 의존성들은 s-파라미터들(S1 내지 S4)에 의해 기술될 수 있다.

[0053]

반복 프로세서(102)는 (추가로) 처리된 채널들을 도출하기 위해 입력 채널들 및/또는 처리된 채널들에 대해 다채널 처리 연산들을 수행하기 위해 스테레오 박스들(110, 112)을 사용할 수 있다. 예를 들어, 반복 프로세서(102)는 일반, 예측 기반 또는 KLT(Karhunen-Loeve-Transformation) 기반 회전 스테레오 박스들(110, 112)을 사용하도록 구성될 수 있다.

[0054]

일반 인코더(또는 인코더 측 스테레오 박스)는 입력 신호들(I1, I2)을 인코딩하여 다음 식을 기초로 출력 신호들(O1, O2)을 얻도록 구성될 수 있다:

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0055]

일반 디코더(또는 디코더 측 스테레오 박스)는 입력 신호들(I1, I2)을 디코딩하여 다음 식을 기초로 출력 신호들(O1, O2)을 얻도록 구성될 수 있다:

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ s_3 & s_4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0057]

[0058] 예측 기반 인코더(또는 인코더 측 스테레오 박스)는 입력 신호들(I1, I2)을 인코딩하여 다음 식을 기초로 출력 신호들(O1, O2)을 얻도록 구성될 수 있다:

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = 0.5 \cdot \begin{bmatrix} 1 & 1 \\ 1-p & -(1+p) \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0059] 여기서, p는 예측 계수이다.

[0061] 예측 기반 디코더(또는 디코더 측 스테레오 박스)는 입력 신호들(I1, I2)을 디코딩하여 다음 식을 기초로 출력 신호들(O1, O2)을 얻도록 구성될 수 있다.

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} 1+p & 1 \\ 1-p & -1 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0063] KLT 기반 회전 인코더(또는 인코더 측 스테레오 박스)는 입력 신호들(I1 내지 I2)을 인코딩하여 다음 식을 기초로 출력 신호들(O1, O2)을 얻도록 구성될 수 있다:

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0065] KLT 기반 회전 디코더(또는 디코더 측 스테레오 박스)는 입력 신호들(I1, I2)을 디코딩하여 다음 식(역회전)에 따라 출력 신호들(O1, O2)을 얻도록 구성될 수 있다:

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

[0067] 다음에, KLT 기반 회전에 대한 회전각( $\alpha$ )의 계산이 설명된다.

[0068] KLT 기반 회전에 대한 회전각( $\alpha$ )은 다음과 같이 정의될 수 있으며:

$$\alpha = \frac{1}{2} \tan^{-1} \left( \frac{2c_{12}}{c_{11}-c_{22}} \right)$$

[0069]  $c_{xy}$ 는 정규화되지 않은 상관 행렬의 항목들이고, 여기서  $c_{11}$ ,  $c_{12}$ 은 채널 에너지들이다.

[0071] 이것은 atan2 함수를 사용하여 분자의 음의 상관들과 분모의 음의 에너지 차이 간의 구별을 가능하게 하도록 구현될 수 있다.

[0072] alpha = 0.5\*atan2(2\*correlation[ch1][ch2],

[0073] (correlation[ch1][ch1] - correlation[ch2][ch2]));

[0074] 또한, 반복 프로세서(102)는 복수의 대역들에 대한 단일 채널 간 상관 값이 얻어지게 복수의 대역들을 포함하는 각각의 채널의 프레임을 사용하여 채널 간 상관을 계산하도록 구성될 수 있으며, 반복 프로세서(102)는 제 1 또는 제 2 다채널 파라미터들이 복수의 대역들 각각으로부터 얻어지도록 복수의 대역들 각각에 대한 다채널 처리를 수행하도록 구성될 수 있다.

[0075] 이로써, 반복 프로세서(102)는 다채널 처리에서 스테레오 파라미터들을 계산하도록 구성될 수 있으며, 반복 프로세서(102)는 스테레오 파라미터가 스테레오 양자화기(예컨대, KLT 기반 회전 인코더)로 정의된, 0으로 양자화된 임계치보다 더 높은 대역들에서만 스테레오 처리를 수행하도록 구성될 수 있다. 스테레오 파라미터들은 예를 들어, MS On/Off 또는 회전각들 또는 예측 계수들일 수 있다.

[0076] 예를 들어, 반복 프로세서(102)는 다채널 처리에서 회전각들을 계산하도록 구성될 수 있으며, 반복 프로세서(102)는 회전각이 회전각 양자화기(예컨대, KLT 기반 회전 인코더)로 정의된, 0으로 양자화된 임계치보다 더 높은 대역들에서만 회전 처리를 수행하도록 구성될 수 있다.

[0077] 따라서 인코더(100)(또는 출력 인터페이스(106))는 완전한 스펙트럼(전대역 박스)에 대한 하나의 파라미터로서 또는 스펙트럼의 부분들에 대한 다수의 주파수 의존 파라미터들로서 변환/회전 정보를 송신하도록 구성될 수 있다.

[0078] 인코더(100)는 다음 표들에 기초하여 비트스트림(107)을 생성하도록 구성될 수 있다:

[0079]

표 1 — mpegh3daExtElementConfig()의 선택스

선택스	비트 수	Mnemonic
<pre>mpegh3daExtElementConfig() {     usacExtElementType      = escapedValue(4, 8, 16);     usacExtElementConfigLength = escapedValue(4, 8, 16);      if (usacExtElementDefaultLengthPresent) {         usacExtElementDefaultLength = escapedValue(8, 16, 0) + 1;     } else {         usacExtElementDefaultLength = 0;     }      usacExtElementPayloadFrag; }  switch (usacExtElementType) { case ID_EXT_ELE_FILL:     /* No configuration element */     break; case ID_EXT_ELE_MPEGS:     SpatialSpecificConfig();     break; case ID_EXT_ELE_SAOC:     SAOCspecificConfig();     break; case ID_EXT_ELE_AUDIOPREROLL:     /* No configuration element */     break; }</pre>	1	uimsbf

[0080]

```

case ID_EXT_ELE_UNI_DRC:
    mpegh3daUniDrcConfig();
    break;
case ID_EXT_ELE_OBJ_METADATA:
    ObjectMetadataConfig();
    break;
case ID_EXT_ELE_SAOC_3D:
    SAOC3DSpecificConfig();
    break;
case ID_EXT_ELE_HOA:
    HOAConfig();
    break;
case ID_EXT_ELE_MCC: /* multi channel coding */
    MCCConfig(grp);
    break;
case ID_EXT_ELE_FMT_CNRVTR
    /* No configuration element */
    break;
default:                                     NOTE
    while (usacExtElementConfigLength--) {
        tmp;                                8          uimsbf
    }
    break;
}
}

```

**NOTE:** usacExtElementType에 대한 디폴트 항목이 알 수 없는 extElementTypes에 사용되어 레거시 디코더들이 향후 확장들에 대처할 수 있다.

[0081]

[0082]

표 2 — MCCConfig()의 선택스

선택스	비트 수	Mnemonic
MCCConfig(grp)		
{		
nChannels = 0		
for(chan=0;chan < bsNumberOfSignals[grp]; chan++)		
chanMask[chan]                                1		
if(chanMask[chan] > 0) {		
mctChannelMap[nChannels]=chan;		
nChannels++;		
}		
}		
}		
<b>NOTE:</b> 해당 ID_USAC_EXT 엘리먼트가 특정 신호 그룹(grp)의 임의의 오디오 엘리먼트에 우선해야 한다.		

[0083]

[0084]

표 3 – MultichannelCodingBoxBandWise()의 선택스

선택스	비트 수	Mnemonic
MultichannelCodingBoxBandWise()		
{		
for(pair=0; pair<numPairs;pair++) {		
if (keepTree == 0) {		
channelPairIndex[pair]	nBits	NOTE 1)
}		
else {		
channelPairIndex[pair]=		
lastChannelPairIndex[pair];		
}		
hasMctMask	1	
hasBandwiseAngles	1	
if (hasMctMask    hasBandwiseAngles) {		
isShort	1	
numMaskBands;	5	
if (isShort) {		
numMaskBands = numMaskBands*8		
}		
} else {		NOTE 2)
numMaskBands = MAX_NUM_MC_BANDS;		
}		
if (hasMctMask) {		
for(j=0;j<numMaskBands;j++) {		

[0085]

```

        msMask[pair][j];           1
    } else {
        for(j=0;j<numMaskBands;j++) {
            msMask[pair][j] = 1;
        }
    }
    if(indepFlag > 0) {
        delta_code_time = 0;
    } else {
        delta_code_time;          1
    }
    if (hasBandwiseAngles == 0) {
        hcod_angle[dpcm_alpha[pair][0]];   1..10      vlcibf
    }
    else {
        for(j=0;j< numMaskBands;j++) {
            if (msMask[pair][j] ==1) {
                hcod_angle[dpcm_alpha[pair][j]];   1..10      vlcibf
            }
        }
    }
}

```

NOTE 1) nBits = floor(log2(nChannels\*(nChannels-1)/2 - 1))+1

[0086]

[0087]

표 4 – MultichannelCodingBoxFullband()의 신택스

신택스	비트 수	Mnemonic
<b>MultichannelCodingBoxFullband()</b>		
{		
for (pair=0; pair<numPairs; pair++) {		
if(keepTree == 0) {		
channelPairIndex[pair]           nBits		
}	NOTE 1)	
else {		
numPairs = lastNumPairs;		
}		
alpha;                            8		
}		
NOTE: 1) nBits = floor(log2(nChannels*(nChannels-1)/2 - 1))+1		

[0088]

[0089]

표 5 — MultichannelCodingFrame()의 선택스

선택스	번호	Mnemonic
MultichannelCodingFrame()		
{		
MCCSignalingType	2	
keepTree	1	
if(keepTree==0) {		
numPairs	5	
}		
else {		
numPairs=lastNumPairs;		
}		
if(MCCSignalingType == 0) /* tree of standard stereo boxes */		
for(i=0;i<numPairs;i++) {		
MCCBox[i] = StereoCoreToolInfo(0);		
}		
}		
if(MCCSignalingType == 1) /* arbitrary mct trees */		
MultichannelCodingBoxBandWise();		
}		
if(MCCSignalingType == 2) /* transmitted trees */		
}		
if(MCCSignalingType == 3) /* simple fullband tree */		
MultichannelCodingBoxFullband();		
}		
}		

[0090]

[ 표 2 ]

[0091]

표 6 — usacExtElementType의 값

usacExtElementType	값
ID_EXT_ELE_FILL	0
ID_EXT_ELE_MPEGS	1
ID_EXT_ELE_SAOC	2
ID_EXT_ELE_AUDIOPREROLL	3
ID_EXT_ELE_UNI_DRC	4
ID_EXT_ELE_OBJ_METADATA	5
ID_EXT_ELE_SAOC_3D	6
ID_EXT_ELE_HOA	7
ID_EXT_ELE_FMT_CNRTR	8
ID_EXT_ELE_MCC	9 또는 10
/* reserved for ISO use */	10-127
/* reserved for use outside of ISO scope */	128 및 그 이상

NOTE: 애플리케이션 특정 usacExtElementType 값들은 ISO 범위 외부에서 사용하도록 예비된 공간에 있도록 지시된다. 이러한 확장들을 스kip하기 위해 디코더에 의해 최소한의 구조가 요구되기 때문에 이들은 디코더에 의해 스kip된다.

[0093]

[ 표 3 ]

[0095] 표 7 - 확장 페이로드 디코딩을 위한 데이터 블록들의 해석

usacExtElementType	연결된 usacExtElementSegmentData는 아래와 같이 나타냄:
ID EXT ELE FILL	Series of fill byte
ID EXT ELE MPEGS	SpatialFrame()
ID EXT ELE SAOC	SaocFrame()
ID EXT ELE AUDIOPREROLL	AudioPreRoll()
ID EXT ELE UNI_DRC	uniDrcGain() as defined in ISO/IEC 23003-4
ID EXT ELE OBJ_METADATA	object metadata()
ID EXT ELE SAOC_3D	Saoc3DFrame()
ID EXT ELE HOA	HOAFrame()
ID EXT ELE FMT_CNRTR	FormatConverterFrame()
ID EXT ELE MCC	MultichannelCodingFrame()
unknown	알려지지 않은 데이터. 데이터 블록이 폐기될 것이다.

[0096]

[0097] 도 3은 일 실시예에 따른 반복 프로세서(102)의 개략적인 블록도를 보여준다. 도 3에 도시된 실시예에서, 다채널 신호(101)는 6개의 채널들: 좌측 채널(L), 우측 채널(R), 좌측 서라운드 채널(Ls), 우측 서라운드 채널(Rs), 센터 채널(C) 및 저주파 효과 채널(LFE)을 갖는 5.1 채널 신호이다.

[0098]

도 3에 나타낸 바와 같이, LFE 채널은 반복 프로세서(102)에 의해 처리되지 않는다. 이는 LFE 채널과 다른 5개의 채널들(L, R, Ls, Rs, Rs) 각각 사이의 채널간 상관 값들이 C가 작기 때문에 또는 채널 마스크가 LFE 채널을 처리하지 않음을 나타내기 때문에 다음과 같이 가정될 경우가 될 수도 있다.

[0099]

제 1 반복 단계에서, 반복 프로세서(102)는 제 1 반복 단계에서 가장 큰 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 5개의 채널들(L, R, Ls, Rs, C)의 각각의 쌍 사이의 채널 간 상관 값을 계산한다. 도 3에서, 반복 프로세서(102)가 제 1 및 제 2 처리된 채널들(P1, P2)을 도출하기 위해, 다채널 처리 연산을 수행하는 스테레오 박스(또는 스테레오 툴)(110)를 사용하여 좌측 채널(L) 및 우측 채널(R)을 처리하도록 좌측 채널(L) 및 우측 채널(R)이 가장 높은 값을 갖는 것으로 가정된다.

[0100]

제 2 반복 단계에서, 반복 프로세서(102)는 제 2 반복 단계에서 가장 큰 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 5개의 채널들(L, R, Ls, Rs, C)의 각각의 쌍과 처리된 채널들(P1, P2) 사이의 채널 간 상관 값을 계산한다. 도 3에서, 반복 프로세서(102)가 제 3 및 제 4 처리된 채널들(P3, P4)을 도출하기 위해, 스테레오 박스(또는 스테레오 툴)(112)를 사용하여 좌측 서라운드 채널(Ls) 및 우측 서라운드 채널(Rs)을 처리하도록 좌측 서라운드 채널(Ls) 및 우측 서라운드 채널(Rs)이 가장 높은 값을 갖는 것으로 가정된다.

[0101]

제 3 반복 단계에서, 반복 프로세서(102)는 제 3 반복 단계에서 가장 큰 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 5개의 채널들(L, R, Ls, Rs, C)의 각각의 쌍과 처리된 채널들(P1 내지 P4) 사이의 채널 간 상관 값을 계산한다. 도 3에서, 반복 프로세서(102)가 제 5 및 제 6 처리된 채널들(P5, P6)을 도출하기 위해, 스테레오 박스(또는 스테레오 툴)(114)를 사용하여 제 1 처리된 채널(P1) 및 제 3 처리된 채널(P3)을 처리하도록 제 1 처리된 채널(P1) 및 제 3 처리된 채널(P3)이 가장 높은 값을 갖는 것으로 가정된다.

[0102]

제 4 반복 단계에서, 반복 프로세서(102)는 제 4 반복 단계에서 가장 큰 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하기 위해 5개의 채널들(L, R, Ls, Rs, C)의 각각의 쌍과 처리된 채널들(P1 내지 P6) 사이의 채널 간 상관 값을 계산한다. 도 3에서, 반복 프로세서(102)가 제 7 및 제 8 처리된 채널들(P7, P8)을 도출하기 위해, 스테레오 박스(또는 스테레오 툴)(116)를 사용하여 제 5 처리된 채널(P5) 및 센터 채널(C)을 처리하도록 제 5 처리된 채널(P5) 및 센터 채널(C)이 가장 높은 값을 갖는 것으로 가정된다.

[0103]

스테레오 박스들(110 내지 116)은 MS 스테레오 박스들, 즉 중간 채널 및 측면 채널을 제공하도록 구성된 중간/측면 입체 음향 박스들일 수 있다. 중간 채널은 스테레오 박스의 입력 채널들의 합일 수 있으며, 측면 채널은 스테레오 박스의 입력 채널들 간의 차이일 수 있다. 또한, 스테레오 박스들(110, 116)은 회전 박스들 또는 스테레오 예측 박스들일 수 있다.

[0104]

도 3에서, 제 1 처리된 채널(P1), 제 3 처리된 채널(P3) 및 제 5 처리된 채널(P5)은 중간 채널들일 수 있으며, 제 2 처리된 채널(P2), 제 4 처리된 채널(P4) 및 제 6 처리된 채널(P6)은 측면 채널들일 수 있다.

[0105]

또한, 도 3에 나타낸 바와 같이, 반복 프로세서(102)는 제 2 반복 단계에서 그리고 적용 가능하다면, 입력 채널들(L, R, Ls, Rs, C)을 그리고 처리된 채널들 중 중간 채널들(P1, P3, P5)(만)을 사용하여 임의의 추가 반복 단계에서 계산, 선택 및 처리를 수행하도록 구성될 수 있다. 즉, 반복 프로세서(102)는 제 2 반복 단계에서 그리

고 적용 가능하다면 임의의 추가 반복 단계에서 계산, 선택 및 처리에 처리된 채널들 중 측면 채널들(P1, P3, P5)을 사용하지 않도록 구성될 수 있다.

[0106] 도 4는 인코딩된 채널들(E1 내지 E3)과 적어도 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 갖는 인코딩된 다채널 신호(107)를 디코딩하기 위한 장치(디코더)(200)의 개략적인 블록도를 도시한다. 이 장치(200)는 채널 디코더(202) 및 다채널 프로세서(204)를 포함한다.

[0107] 채널 디코더(202)는 D1 내지 D3에서 디코딩된 채널들을 얻기 위해, 인코딩된 채널들(E1 내지 E3)을 디코딩하도록 구성된다.

[0108] 예를 들어, 채널 디코더(202)는 적어도 3개의 모노 디코더들(또는 모노 박스들 또는 모노 틀들)(206\_1 내지 206\_3)을 포함할 수 있으며, 모노 디코더들(206\_1 내지 206\_3) 각각은 적어도 3개의 인코딩된 채널들(E1 내지 E3)을 디코딩하여 각각의 디코딩된 채널(D1 내지 D3)을 얻도록 구성될 수 있다. 모노 디코더들(206\_1 내지 206\_3)은 예를 들어 변환 기반 오디오 디코더들일 수 있다.

[0109] 다채널 프로세서(204)는 처리된 채널들을 얻기 위해 제 2 다채널 파라미터들(MCH\_PAR2)에 의해 식별된 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 제 2 다채널 파라미터들(MCH\_PAR2)을 사용하여 다채널 처리를 수행하고, 제 1 다채널 파라미터들(MCH\_PAR1)에 의해 식별된 채널들의 제 1 쌍을 사용하여 그리고 제 1 다채널 파라미터들(MCH\_PAR1)을 사용하여 추가 다채널 처리를 수행하도록 구성되며, 여기서 채널들의 제 1 쌍은 적어도 하나의 처리된 채널을 포함한다.

[0110] 예로서 도 4에 나타낸 바와 같이, 제 2 다채널 파라미터들(MCH\_PAR2)은 디코딩된 채널들의 제 2 쌍이 제 1 디코딩된 채널(D1) 및 제 2 디코딩된 채널(D2)로 구성된다는 것을 표시(시그널링)할 수 있다. 따라서 다채널 프로세서(204)는 처리된 채널들(P1\*, P2\*)을 얻기 위해 (제 2 다채널 파라미터들(MCH\_PAR2)에 의해 식별된) 제 1 디코딩된 채널(D1) 및 제 2 디코딩된 채널(D2)로 구성된 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 제 2 다채널 파라미터들(MCH\_PAR2)을 사용하여 다채널 처리를 수행한다. 제 1 다채널 파라미터들(MCH\_PAR1)은 디코딩된 채널들의 제 1 쌍이 제 1 처리된 채널(P1\*) 및 제 3 디코딩된 채널(D3)로 구성됨을 나타낼 수 있다. 따라서 다채널 프로세서(204)는 처리된 채널들(P3\*, P4\*)을 획득하기 위해, (제 1 다채널 파라미터들(MCH\_PAR1)에 의해 식별된) 제 1 처리된 채널(P1\*) 및 제 3 디코딩된 채널(D3)로 구성된 디코딩된 채널들의 제 1 쌍을 사용하여 그리고 제 1 다채널 파라미터들(MCH\_PAR1)을 사용하여 추가 다채널 처리를 수행한다.

[0111] 또한, 다채널 프로세서(204)는 제 3 처리된 채널(P3\*)을 제 1 채널(CH1)로서, 제 4 처리된 채널(P4\*)을 제 3 채널(CH3)로서 그리고 제 2 처리된 채널(P2\*)을 제 2 채널(CH2)로서 제공할 수 있다.

[0112] 도 4에 도시된 디코더(200)가 도 1에 도시된 인코더(100)로부터 인코딩된 다채널 신호(107)를 수신한다고 가정하면, 디코더(200)의 제 1 디코딩된 채널(D1)은 인코더(100)의 제 3 처리된 채널(P3)과 동등할 수 있고, 디코더(200)의 제 2 디코딩된 채널(D2)은 인코더(100)의 제 4 처리된 채널(P4)과 동등할 수 있으며, 디코더(200)의 제 3 디코딩된 채널(D3)은 인코더(100)의 제 2 처리된 채널(P2)과 동등할 수 있다. 또한, 디코더(200)의 제 1 처리된 채널(P1\*)은 인코더(100)의 제 1 처리된 채널(P1)과 동등할 수 있다.

[0113] 또한, 인코딩된 다채널 신호(107)는 직렬 신호일 수 있으며, 여기서 제 2 다채널 파라미터들(MCH\_PAR2)은 제 1 다채널 파라미터들(MCH\_PAR1) 전에 디코더(200)에서 수신된다. 그 경우, 다채널 프로세서(204)는 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)이 디코더에 의해 수신되는 순서로 디코딩된 채널들을 처리하도록 구성될 수 있다. 도 4에 도시된 예에서, 디코더는 제 1 다채널 파라미터들(MCH\_PAR1) 전에 제 2 다채널 파라미터들(MCH\_PAR2)을 수신하고, 이에 따라 제 1 다채널 파라미터(MCH\_PAR1)에 의해 식별된 (제 1 처리된 채널(P1\*) 및 제 3 디코딩된 채널(D3)로 구성된) 디코딩된 채널들의 제 1 쌍을 사용하여 다채널 처리를 수행하기 전에 제 2 다채널 파라미터(MCH\_PAR2)에 의해 식별된 (제 1 및 제 2 디코딩된 채널들(D1, D2)로 구성된) 디코딩된 채널들의 제 2 쌍을 사용하여 다채널 처리를 수행한다.

[0114] 도 4에서, 다채널 프로세서(204)는 예시적으로 2개의 다채널 처리 연산들을 수행한다. 설명의 목적으로, 다채널 프로세서(204)에 의해 수행되는 다채널 처리 연산들은 도 4에 처리 박스들(208, 210)로 예시된다. 처리 박스들(208, 210)은 하드웨어 또는 소프트웨어로 구현될 수 있다. 처리 박스들(208, 210)은 예를 들어, 인코더(100)와 관련하여 앞서 논의한 바와 같은 스테레오 박스들, 이를테면 일반적인 디코더들(또는 디코더 측 스테레오 박스들), 예측 기반 디코더들(또는 디코더 측 스테레오 박스들) 또는 KLT 기반 회전 디코더들(또는 디코더 측 스테레오 박스들)일 수 있다.

[0115] 예를 들어, 인코더(100)는 KLT 기반 회전 인코더들(또는 인코더 측 스테레오 박스들)을 사용할 수 있다. 그 경

우, 인코더(100)는 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)이 회전각들을 갖도록 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 도출할 수 있다. 회전각들은 차동적으로 인코딩될 수 있다. 따라서 디코더(200)의 다채널 프로세서(204)는 차동적으로 인코딩된 회전각들을 차동적으로 디코딩하기 위한 차동 디코더를 포함할 수 있다.

[0116] 장치(200)는 인코딩된 다채널 신호(107)를 수신하고 처리하여, 인코딩된 채널들(E1 내지 E3)을 채널 디코더(202)에 그리고 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 다채널 프로세서(204)에 제공하도록 구성된 입력 인터페이스(212)를 더 포함할 수 있다.

[0117] 이미 언급한 바와 같이, 유지 표시자(또는 유지 트리 플래그)는 어떠한 새로운 트리도 송신되지 않지만 마지막 스테레오 트리가 사용될 것임을 시그널링하는데 사용될 수 있다. 이것은 채널 상관 특성들이 더 오랜 시간 동안 고정된 상태라면 동일한 스테레오 트리 구성의 다중 송신을 피하는 데 사용될 수 있다.

[0118] 따라서 인코딩된 다채널 신호(107)가 제 1 프레임에 대해 제 1 또는 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)을 포함하고, 제 1 프레임 다음의 제 2 프레임에 대해서는 유지 표시자를 포함하는 경우, 다채널 프로세서(204)는 제 1 프레임에 사용된 것과 동일한 제 2 쌍 또는 동일한 제 1 쌍의 채널들에 대해 제 2 프레임에서 다채널 처리 또는 추가 다채널 처리를 수행하도록 구성될 수 있다.

[0119] 다채널 처리 및 추가 다채널 처리는 스테레오 파라미터를 사용하는 스테레오 처리를 포함할 수 있는데, 여기서는 디코딩된 채널들(D1 내지 D3)의 개개의 스케일 팩터 대역들 또는 스케일 팩터 대역들의 그룹들에 대해, 제 1 스테레오 파라미터가 제 1 다채널 파라미터(MCH\_PAR1)에 포함되고 제 2 스테레오 파라미터는 제 2 다채널 파라미터(MCH\_PAR2)에 포함된다. 이로써, 제 1 스테레오 파라미터와 제 2 스테레오 파라미터는 회전각들 또는 예측 계수들과 같은 동일한 타입일 수 있다. 당연히, 제 1 스테레오 파라미터와 제 2 스테레오 파라미터는 서로 다른 타입들일 수 있다. 예를 들어, 제 1 스테레오 파라미터는 회전각일 수 있고, 제 2 스테레오 파라미터는 예측 계수일 수 있으며, 또는 그 반대일 수 있다.

[0120] 또한, 제 1 또는 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 어느 스케일 팩터 대역들이 다채널 처리되고 어떤 스케일 팩터 대역들이 다채널 처리되지 않는지를 나타내는 다채널 처리 마스크를 포함할 수 있다. 이로써, 다채널 프로세서(204)는 다채널 처리 마스크에 의해 표시된 스케일 팩터 대역들에서 다채널 처리를 수행하지 않도록 구성될 수 있다.

[0121] 제 1 및 제 2 다채널 파라미터들(MCH\_PAR1, MCH\_PAR2)은 각각 채널 쌍 식별(또는 인덱스)을 포함할 수 있으며, 여기서 다채널 프로세서(204)는 미리 정의된 디코딩 규칙 또는 인코딩된 다채널 신호에서 표시된 디코딩 규칙을 사용하여 채널 쌍 식별들(또는 인덱스들)을 디코딩하도록 구성될 수 있다.

[0122] 예를 들어, 인코더(100)와 관련하여 앞서 설명한 바와 같이, 채널 쌍들은 총 채널 수에 따라 각각의 쌍에 대한 고유 인덱스를 사용하여 효율적으로 시그널링될 수 있다.

[0123] 또한, 디코딩 규칙은 허프만 디코딩 규칙일 수 있으며, 여기서 다채널 프로세서(204)는 채널 쌍 식별들의 허프만 디코딩을 수행하도록 구성될 수 있다.

[0124] 인코딩된 다채널 신호(107)는 다채널 처리가 허용되는 디코딩된 채널들의 하위 그룹만을 표시하고 다채널 처리가 허용되지 않는 적어도 하나의 디코딩된 채널을 표시하는 다채널 처리 허용 표시자를 더 포함할 수 있다. 이로써, 다채널 프로세서(204)는 다채널 처리 허용 표시자에 의해 표시된 바와 같이 다채널 처리가 허용되지 않는 적어도 하나의 디코딩된 채널에 대해 어떠한 다채널 처리도 수행하지 않도록 구성될 수 있다.

[0125] 예를 들어, 다채널 신호가 5.1 채널 신호인 경우, 다채널 처리 허용 표시자는 다채널 처리가 5개의 채널들, 즉 우측(R), 좌측(L), 우측 서라운드(Rs), 좌측 서라운드(Ls) 및 센터(C)에만 허용됨을 나타낼 수 있고, 여기서 LFE 채널에는 다채널 처리가 허용되지 않는다.

[0126] 디코딩 프로세스(채널 쌍 인덱스들의 디코딩)의 경우, 다음과 같은 c 코드가 사용될 수 있다. 이로써, 모든 채널 쌍들에 대해, 현재 프레임의 채널 쌍들의 수(numPairs)뿐만 아니라 활성 KLT 처리에 의한 채널들의 수(nChannels)도 필요하다.

```

maxNumPairIdx = nChannels*(nChannels-1)/2 - 1;
numBits = floor(log2(maxNumPairIdx))+1;
pairCounter = 0;

for (chanl=1; chanl < nChannels; chanl++) {

    for (chan0=0; chan0 < chanl; chan0++) {
        if (pairCounter == pairIdx) {
            channelPair[0] = chan0;
            channelPair[1] = chanl;
            return;
        }
        else
            pairCounter++;
    }
}

```

[0127]

비-대역 방식의 각들에 대한 예측 계수들을 디코딩하기 위해, 다음과 같은 c 코드가 사용될 수 있다.

```

for(pair=0; pair<numPairs; pair++) {
    mctBandsPerWindow = numMaskBands[pair]/windowsPerFrame;

    if(delta_code_time[pair] > 0) {
        lastVal = alpha_prev_fullband[pair];
    } else {
        lastVal = DEFAULT_ALPHA;
    }

    newAlpha = lastVal + dpcm_alpha[pair][0];
    if(newAlpha >= 64) {
        newAlpha -= 64;
    }

    for (band=0; band < numMaskBands; band++){
        /* set all angles to fullband angle */
        pairAlpha[pair][band] = newAlpha;

        /* set previous angles according to mctMask */
        if(mctMask[pair][band] > 0) {
            alpha_prev_frame[pair][band%mctBandsPerWindow] = newAlpha;
        } else {
            alpha_prev_frame[pair][band%mctBandsPerWindow] = DEFAULT_ALPHA;
        }
    }
    alpha_prev_fullband[pair] = newAlpha;
    for(band=bandsPerWindow ; band<MAX_NUM_MC_BANDS; band++) {
        alpha_prev_frame[pair][band] = DEFAULT_ALPHA;
    }
}

```

[0129]

비-대역 방식의 KLT 각들에 대한 예측 계수들을 디코딩하기 위해, 다음과 같은 c 코드가 사용될 수 있다.

```

for(pair=0; pair<numPairs; pair++) {
    mctBandsPerWindow = numMaskBands[pair]/windowsPerFrame;
    for(band=0; band<numMaskBands[pair]; band++) {

```

[0131]

```

    if(delta_code_time[pair] > 0) {
        lastVal = alpha_prev_frame[pair][band%mctBandsPerWindow];
    }
    else {
        if ((band % mctBandsPerWindow) == 0) {
            lastVal = DEFAULT_ALPHA;
        }
    }
    if (msMask[pair][band] > 0 ) {

        newAlpha = lastVal + dpcm_alpha[pair][band];
        if(newAlpha >= 64) {
            newAlpha -= 64;
        }
        pairAlpha[pair][band] = newAlpha;
        alpha_prev_frame[pair][band%mctBandsPerWindow] = newAlpha;
        lastVal = newAlpha;
    }
    else {
        alpha_prev_frame[pair][band%mctBandsPerWindow] = DEFAULT_ALPHA; /* -45° */
    }
}

/* reset fullband angle */
alpha_prev_fullband[pair] = DEFAULT_ALPHA;
}
for(band=bandsPerWindow ; band<MAX_NUM_MC_BANDS; band++) {
    alpha_prev_frame[pair][band] = DEFAULT_ALPHA;
}
}

```

[0132]

[0133]

서로 다른 플랫폼들에서 삼각 함수들의 부동 소수점 차이들을 피하기 위해, 각도 인덱스들을 sin/cos으로 직접 변환하기 위한 다음의 검색 표들이 사용될 것이다:

```

tabIndexToSinAlpha[64] = {
-1.000000f, -0.998795f, -0.995185f, -0.989177f, -0.980785f, -0.970031f, -
0.956940f, -0.941544f,
-0.923880f, -0.903989f, -0.881921f, -0.857729f, -0.831470f, -0.803208f, -
0.773010f, -0.740951f,
-0.707107f, -0.671559f, -0.634393f, -0.595699f, -0.555570f, -0.514103f,
0.471397f, -0.427555f,
-0.382683f, -0.336890f, -0.290285f, -0.242980f, -0.195090f, -0.146730f, -
0.098017f, -0.049068f,
0.000000f, 0.049068f, 0.098017f, 0.146730f, 0.195090f, 0.242980f,
0.290285f, 0.336890f,
0.382683f, 0.427555f, 0.471397f, 0.514103f, 0.555570f, 0.595699f,
0.634393f, 0.671559f,
0.707107f, 0.740951f, 0.773010f, 0.803208f, 0.831470f, 0.857729f,
0.881921f, 0.903989f,
0.923880f, 0.941544f, 0.956940f, 0.970031f, 0.980785f, 0.989177f,
0.995185f, 0.998795f
};

```

[0134]

```

tabIndexToCosAlpha[64] = {

0.000000f, 0.049068f, 0.098017f, 0.146730f, 0.195090f, 0.242980f,
0.290285f, 0.336890f,
0.382683f, 0.427555f, 0.471397f, 0.514103f, 0.555570f, 0.595699f,
0.634393f, 0.671559f,
0.707107f, 0.740951f, 0.773010f, 0.803208f, 0.831470f, 0.857729f,
0.881921f, 0.903989f,
0.923880f, 0.941544f, 0.956940f, 0.970031f, 0.980785f, 0.989177f,
0.995185f, 0.998795f,
1.000000f, 0.998795f, 0.995185f, 0.989177f, 0.980785f, 0.970031f,
0.956940f, 0.941544f,
0.923880f, 0.903989f, 0.881921f, 0.857729f, 0.831470f, 0.803208f,
0.773010f, 0.740951f,
0.707107f, 0.671559f, 0.634393f, 0.595699f, 0.555570f, 0.514103f,
0.471397f, 0.427555f,
0.382683f, 0.336890f, 0.290285f, 0.242980f, 0.195090f, 0.146730f,
0.098017f, 0.049068f
};

```

[0135]

[0136]

다채널 코딩의 디코딩을 위해, 다음의 c 코드가 KLT 회전 기반 접근 방식에 사용될 수 있다.

```
decode_mct_rotation()
{
    for (pair=0; pair < self->numPairs; pair++) {
        mctBandOffset = 0;

        /* inverse MCT rotation */
        for (win = 0, group = 0; group < num_window_groups; group++) {

            for (groupwin = 0; groupwin < window_group_length[group]; groupwin++, win++) {
                *dmx = spectral_data[ch1][win];
                *res = spectral_data[ch2][win];
                apply_mct_rotation_wrapper(self, dmx, res, &alphaSfb[mctBandOffset],
                                           &mctMask[mctBandOffset], mctBandsPerWindow, alpha,
                                           totalSfb, pair, nSamples);
            }

            mctBandOffset += mctBandsPerWindow;
        }
    }
}
```

[0137]

대역 방식의 처리를 위해, 다음과 같은 c 코드가 사용될 수 있다.

```
apply_mct_rotation_wrapper(self, *dmx, *res, *alphaSfb, *mctMask,
                           mctBandsPerWindow,
                           alpha, totalSfb, pair, nSamples)
{
    sfb = 0;

    if (self->MCCSignalingType == 0) {
    }
    else if (self->MCCSignalingType == 1) {
        /* apply fullband box */

        if (!self->bHasBandwiseAngles[pair] && !self->bHasMctMask[pair]) {
            apply_mct_rotation(dmx, res, alphaSfb[0], nSamples);
        }
        else {
            /* apply bandwise processing */
            for (i = 0; i < mctBandsPerWindow; i++) {
                if (mctMask[i] == 1) {
                    startLine = swb_offset[sfb];
                    stopLine = (sfb+2<totalSfb)? swb_offset[sfb+2] : swb_offset[sfb+1];
                    nSamples = stopLine-startLine;

                    apply_mct_rotation(&dmx[startLine], &res[startLine],
                                       alphaSfb[i], nSamples);
                }
                sfb += 2;

                /* break condition */
                if (sfb >= totalSfb) {
                    break;
                }
            }
        }
    }
    else if (self->MCCSignalingType == 2) {
    }
    else if (self->MCCSignalingType == 3) {
        apply_mct_rotation(dmx, res, alpha, nSamples);
    }
}
```

[0140]

[0141] KLT 회전의 적용을 위해, 다음과 같은 c 코드가 사용될 수 있다.

```
apply_mct_rotation(*dmx, *res, alpha, nSamples)
{
    for (n=0;n<nSamples;n++) {
        L = dmx[n] * tabIndexToCosAlpha [alphaIdx] - res[n] *
tabIndexToSinAlpha [alphaIdx];
        R = dmx[n] * tabIndexToSinAlpha [alphaIdx] + res[n] *
tabIndexToCosAlpha [alphaIdx];

        dmx[n] = L;
        res[n] = R;
    }
}
```

[0142]

[0143] 도 5는 적어도 3개의 채널들을 갖는 다채널 신호를 인코딩하기 위한 방법(300)의 흐름도를 도시한다. 이 방법(300)은 제 1 반복 단계에서 적어도 3개의 채널들의 각각의 쌍 사이의 채널 간 상관 값들을 계산하여, 제 1 반복 단계에서 가장 높은 값을 갖거나 임계치보다 큰 값을 갖는 쌍을 선택하고, 선택된 쌍을 다채널 처리 연산을 사용하여 처리하여 선택된 쌍에 대한 제 1 다채널 파라미터들을 도출하고 제 1 처리된 채널들을 도출하는 단계(302); 제 2 다채널 파라미터들 및 제 2 처리된 채널들을 도출하기 위해, 처리된 채널들 중 적어도 하나를 사용하여 제 2 반복 단계에서 계산, 선택 및 처리를 수행하는 단계(304); 인코딩된 채널들을 획득하기 위해, 반복 프로세서에 의해 수행되는 반복 처리로부터 야기되는 채널들을 인코딩하는 단계(306); 및 인코딩된 채널들과 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 생성하는 단계(308)를 포함한다.

[0144]

도 6은 인코딩된 채널들과 적어도 제 1 및 제 2 다채널 파라미터들을 갖는 인코딩된 다채널 신호를 디코딩하기 위한 방법(400)의 흐름도를 도시한다. 이 방법(400)은 디코딩된 채널들을 획득하기 위해, 인코딩된 채널들을 디코딩하는 단계(402); 및 처리된 채널들을 얻기 위해 제 2 다채널 파라미터들에 의해 식별된 디코딩된 채널들의 제 2 쌍을 사용하여 그리고 제 2 다채널 파라미터들을 사용하여 다채널 처리를 수행하고, 제 1 다채널 파라미터들에 의해 식별된 채널들의 제 1 쌍을 사용하여 그리고 제 1 다채널 파라미터들을 사용하여 추가 다채널 처리를 수행하는 단계(404)를 포함하며, 여기서 채널들의 제 1 쌍은 적어도 하나의 처리된 채널을 포함한다.

[0145]

본 발명은 블록들이 실제 또는 논리적 하드웨어 컴포넌트들을 표현하는 블록도들과 관련하여 설명되었지만, 본 발명은 또한 컴퓨터 구현 방법에 의해 구현될 수 있다. 후자의 경우, 블록들은 대응하는 방법 단계들을 나타내는데, 여기서 이러한 단계들은 대응하는 논리적 또는 물리적 하드웨어 블록들에 의해 수행되는 기능들을 의미한다.

[0146]

일부 양상들은 장치와 관련하여 설명되었지만, 이러한 양상들은 또한 대응하는 방법의 설명을 나타내며, 여기서 블록 또는 디바이스는 방법 단계 또는 방법 단계의 특징에 대응한다는 점이 명백하다. 비슷하게, 방법 단계와 관련하여 설명한 양상들은 또한 대응하는 장치의 대응하는 블록 또는 항목 또는 특징의 설명을 나타낸다. 방법 단계들의 일부 또는 전부가 예를 들어, 마이크로프로세서, 프로그래밍 가능한 컴퓨터 또는 전자 회로와 같은 하드웨어 장치에 의해(또는 사용하여) 실행될 수도 있다. 일부 실시예들에서, 가장 중요한 방법 단계들 중 어떤 하나 이상의 단계들이 이러한 장치에 의해 실행될 수도 있다.

[0147]

본 발명의 송신된 또는 인코딩된 신호는 디지털 저장 매체 상에 저장될 수 있고 또는 송신 매체, 예컨대 무선 송신 매체 또는 유선 송신 매체, 예컨대 인터넷을 통해 송신될 수 있다.

[0148]

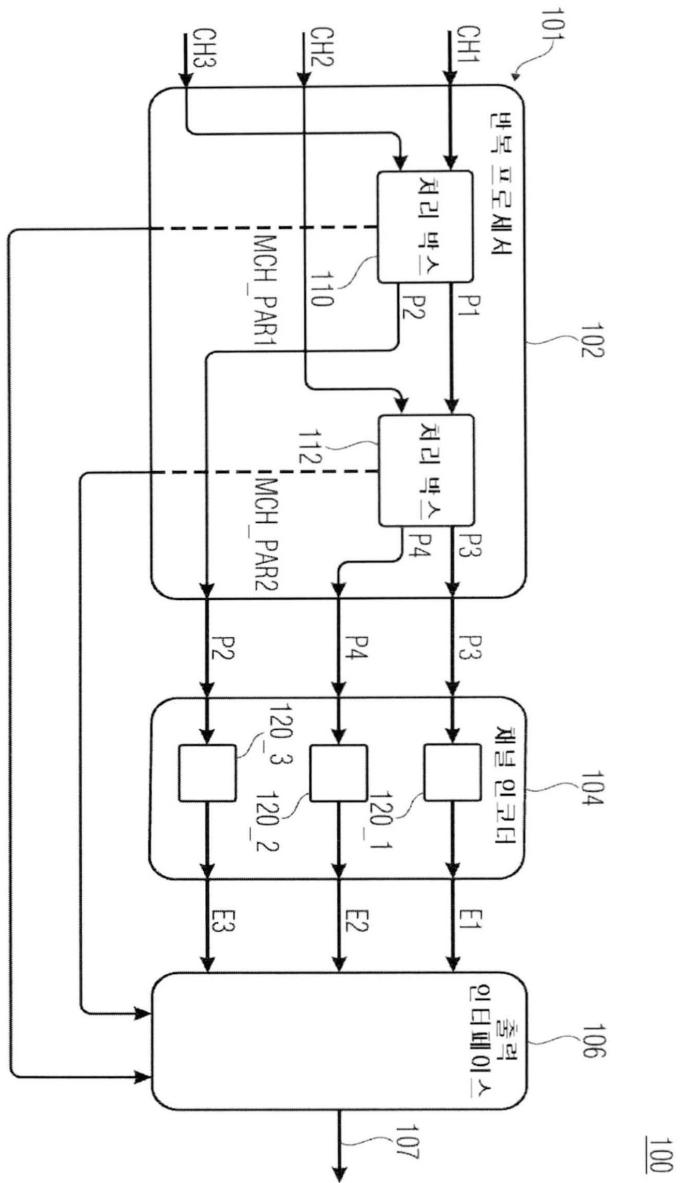
특정 구현 요건들에 따라, 본 발명의 실시예들은 하드웨어로 또는 소프트웨어로 구현될 수 있다. 구현은 각각의 방법이 수행되도록 프로그래밍 가능 컴퓨터 시스템과 협력하는(또는 협력할 수 있는) 전자적으로 판독 가능 제어 신호들이 저장된 디지털 저장 매체, 예를 들어 플로피 디스크, DVD, 블루레이, CD, ROM, PROM 및 EEPROM, EEPROM 또는 플래시 메모리를 사용하여 수행될 수 있다. 따라서 디지털 저장 매체는 컴퓨터 판독 가능할 수도

있다.

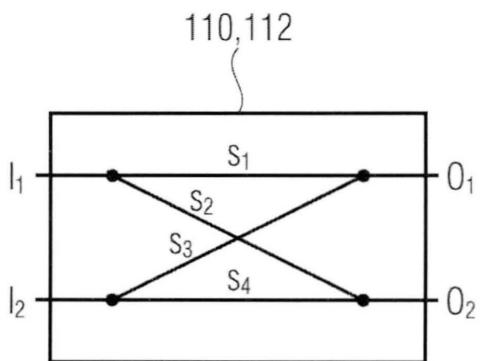
- [0149] 본 발명에 따른 일부 실시예들은 본 명세서에서 설명한 방법들 중 하나가 수행되도록, 프로그래밍 가능 컴퓨터 시스템과 협력할 수 있는 전자적으로 판독 가능 제어 신호들을 갖는 데이터 반송파를 포함한다.
- [0150] 일반적으로, 본 발명의 실시예들은 컴퓨터 프로그램 제품이 컴퓨터 상에서 실행될 때, 방법들 중 하나를 수행하기 위해 작동하는 프로그램 코드를 갖는 컴퓨터 프로그램 제품으로서 구현될 수 있다. 프로그램 코드는 예를 들어, 기계 판독 가능 반송파 상에 저장될 수 있다.
- [0151] 다른 실시예들은 기계 판독 가능 반송파 상에 저장된, 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 컴퓨터 프로그램을 포함한다.
- [0152] 즉, 본 발명의 방법의 한 실시예는 이에 따라, 컴퓨터 상에서 컴퓨터 프로그램이 실행될 때 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 프로그램 코드를 갖는 컴퓨터 프로그램이다.
- [0153] 따라서 본 발명의 방법의 추가 실시예는 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 컴퓨터 프로그램을 포함하여 그 위에 기록된 데이터 반송파(또는 디지털 저장 매체와 같은 비-일시적 저장 매체, 또는 컴퓨터 판독 가능 매체)이다. 데이터 반송파, 디지털 저장 매체 또는 레코딩된 매체는 통상적으로 유형적이고 그리고/ 또는 비-일시적이다.
- [0154] 따라서 본 발명의 방법의 추가 실시예는 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 컴퓨터 프로그램을 나타내는 신호들의 데이터 스트림 또는 시퀀스이다. 신호들의 데이터 스트림 또는 시퀀스는 예를 들어, 데이터 통신 접속을 통해, 예를 들어 인터넷을 통해 전송되도록 구성될 수 있다.
- [0155] 추가 실시예는 처리 수단, 예를 들어 본 명세서에서 설명한 방법들 중 하나를 수행하도록 구성 또는 적응된 컴퓨터 또는 프로그래밍 가능 로직 디바이스를 포함한다.
- [0156] 추가 실시예는 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 컴퓨터 프로그램이 설치된 컴퓨터를 포함한다.
- [0157] 본 발명에 따른 추가 실시예는 본 명세서에서 설명한 방법들 중 하나를 수행하기 위한 컴퓨터 프로그램을 수신기에(예를 들어, 전자적으로 또는 광학적으로) 전송하도록 구성된 장치 또는 시스템을 포함한다. 수신기는 예를 들어, 컴퓨터, 모바일 디바이스, 메모리 디바이스 등일 수도 있다. 장치 또는 시스템은 예를 들어, 컴퓨터 프로그램을 수신기에 전송하기 위한 파일 서버를 포함할 수도 있다.
- [0158] 일부 실시예들에서, 프로그래밍 가능 로직 디바이스(예를 들어, 필드 프로그래밍 가능 게이트 어레이)는 본 명세서에서 설명한 방법들의 기능들 중 일부 또는 전부를 수행하는데 사용될 수 있다. 일부 실시예들에서, 필드 프로그래밍 가능 게이트 어레이는 본 명세서에서 설명한 방법들 중 하나를 수행하기 위해 마이크로프로세서와 협력할 수 있다. 일반적으로, 방법들은 바람직하게 임의의 하드웨어 장치에 의해 수행된다.
- [0159] 앞서 설명한 실시예들은 단지 본 발명의 원리들에 대한 예시일 뿐이다. 본 명세서에서 설명한 배열들 및 세부사항들의 수정들 및 변형들이 다른 당업자들에게 명백할 것이라고 이해된다. 따라서 이는 본 명세서의 실시예들의 묘사 및 설명에 의해 제시된 특정 세부사항들로가 아닌, 첨부된 특허청구범위로만 한정되는 것을 취지로 한다.

도면

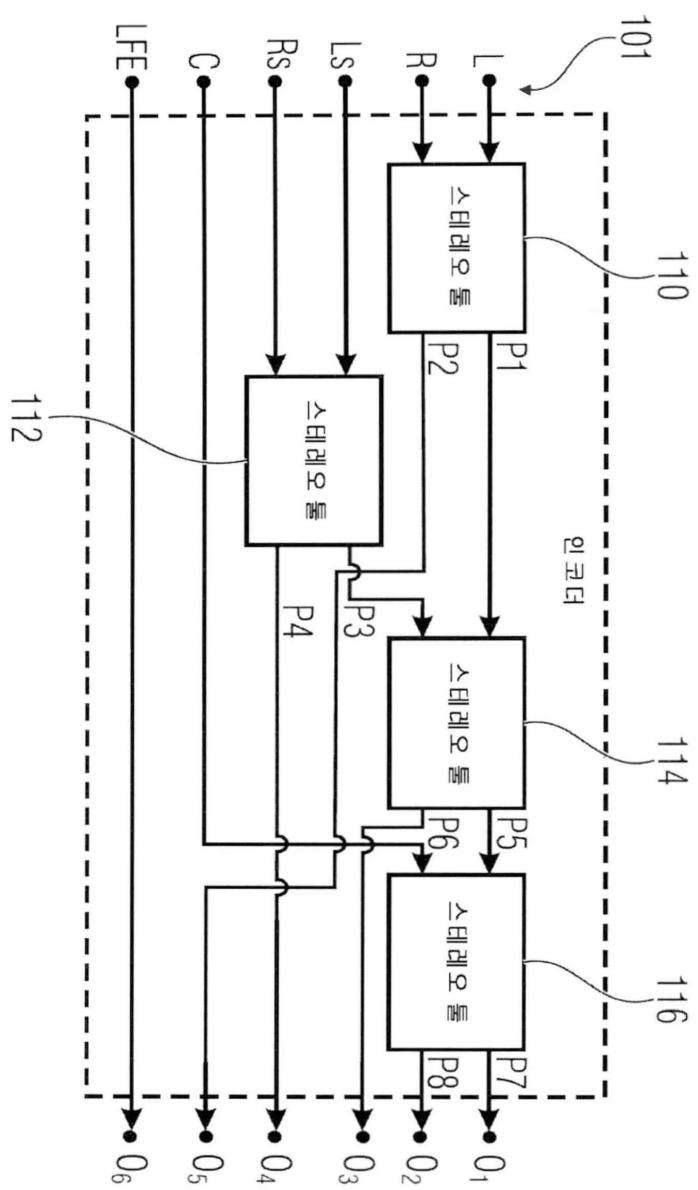
## 도면1



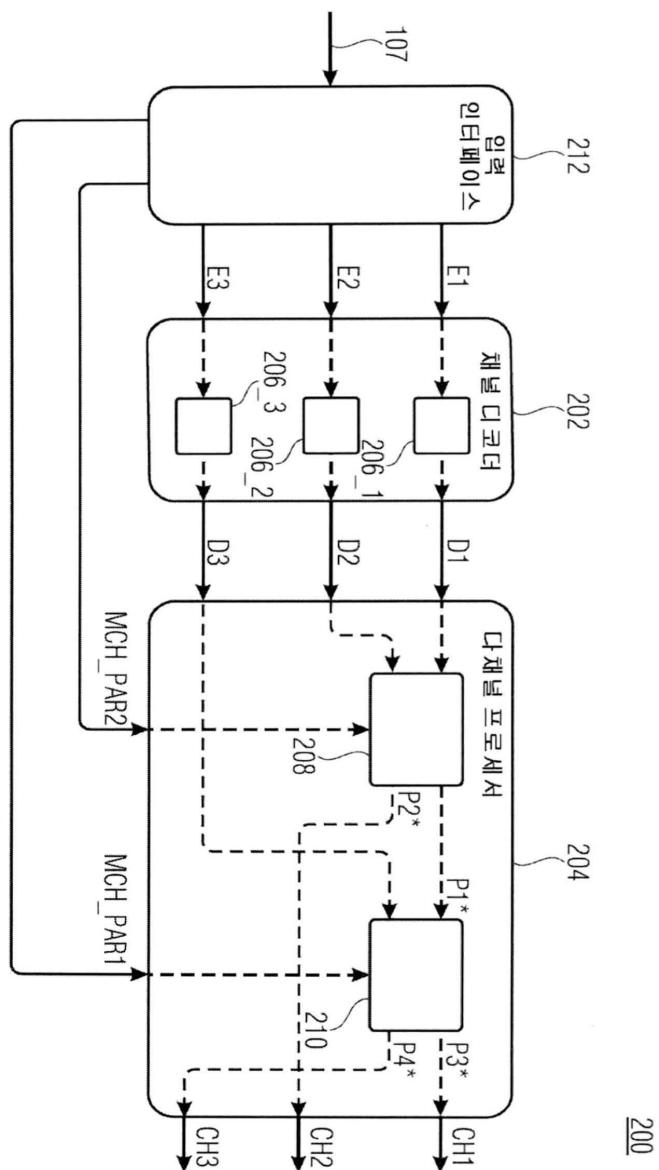
도면2



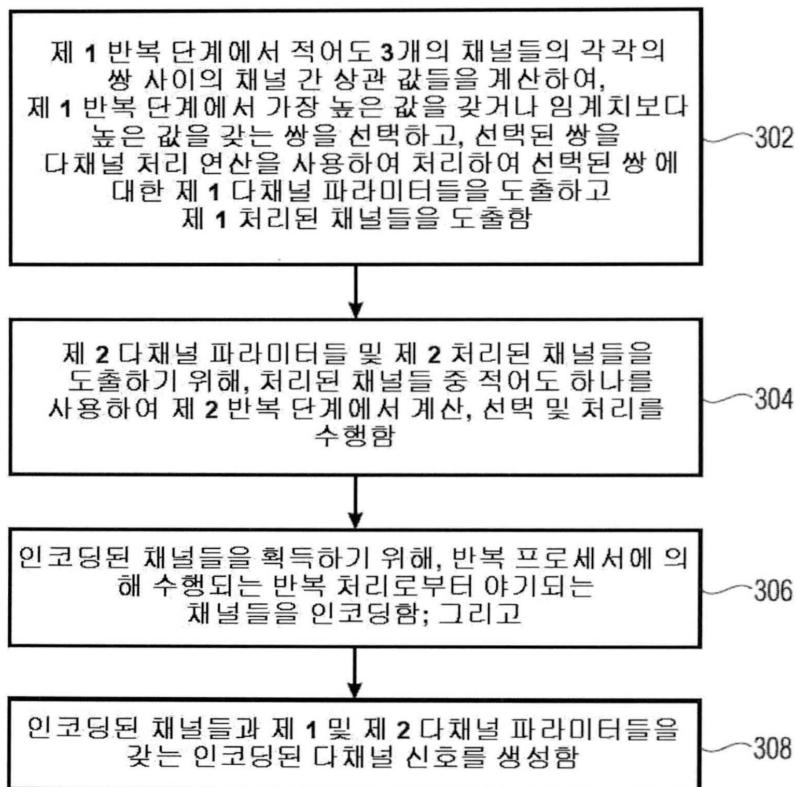
도면3



도면4



## 도면5

300

## 도면6

400