



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2016년11월23일  
 (11) 등록번호 10-1679083  
 (24) 등록일자 2016년11월17일

(51) 국제특허분류(Int. Cl.)  
*G10L 19/00* (2006.01) *H03M 7/30* (2006.01)  
 (21) 출원번호 10-2010-7026245  
 (22) 출원일자(국제) 2009년05월22일  
 심사청구일자 2014년04월28일  
 (85) 번역문제출일자 2010년11월23일  
 (65) 공개번호 10-2011-0021803  
 (43) 공개일자 2011년03월04일  
 (86) 국제출원번호 PCT/US2009/044983  
 (87) 국제공개번호 WO 2009/148858  
 국제공개일자 2009년12월10일  
 (30) 우선권주장  
 12/130,862 2008년05월30일 미국(US)  
 (56) 선행기술조사문헌  
 R. Geiger et al. IntMDCT-A link between  
 perceptual and lossless audio coding, ICASSP,  
 orlando, 13 May 2002, pp.1813-1816\*  
 \*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
**마이크로소프트 테크놀로지 라이선싱, 엘엘씨**  
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
 마이크로소프트 웨이  
 (72) 발명자  
**메흐로트라, 산제브**  
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
 소프트 웨이 마이크로소프트 코퍼레이션 국제 특  
 허부 내  
 (74) 대리인  
**김태홍**

전체 청구항 수 : 총 20 항

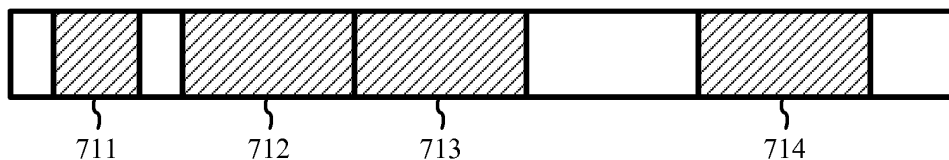
심사관 : 옥윤철

(54) 발명의 명칭 **2개의 블록 변환으로의 중첩 변환의 분해**

**(57) 요약**

오디오 인코더/디코더는 겹침 변환의 가역적인 정수-정수 형태를 제공하기 위해 가역적인 구현을 갖는 중첩 원도  
 우 변환과 블록 변환의 조합을 사용한다. 가역적 겹침 변환은 가변 서브프레임 크기를 갖는 오디오 신호의 손실  
 및 무손실 변환 영역 코딩을 허용한다.

**대표도** - 도7



명세서

청구범위

청구항 1

복수의 상이한 크기의 서브프레임들로 분할된 입력 오디오 신호를 처리하기 위해 겹침 변환(lapped transform)의 가역적 정수 입력/정수 출력 구현들을 도출하는 방법에 있어서,

상기 겹침 변환을, 인접한 서브프레임들을 중첩시키는 중첩 영역에 대한 중첩 윈도우 변환들과 상기 서브프레임들에 대한 블록 변환들의 조합들로 분해하는(factorizing) 단계로서, 상기 겹침 변환을 분해하는 단계는,

$$\begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$

주어진 서브프레임에 대한 상기 겹침 변환의 표현을 형태를 갖는 직사각형 선형 변환 행렬 A - 상기 행렬 A 내의 각 서브블록은 상기 서브프레임의 크기가 N인 경우에 N/2 x N/2 행렬이고, i, j는 상기 행렬 서브블록들의 지수들이며, 상기 행렬 A는  $A_{i(j-1)}A_{(i+1)(j-1)}^{-1} = A_{ij}A_{(i+1)j}^{-1}$  라는 속성을 가진 - 로서 형성하는 단계; 및

$$\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$$

상기 직사각형 선형 변환 행렬 A를 형태의 중첩 윈도우 변환 행렬 C와

$$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$$

형태의 블록 변환 행렬 B로 분해하는 단계를 포함하는, 상기 분해하는 단계와;

오디오 프로세서에서, 상기 서브프레임을 갖는 오디오 신호의 입력을 수신하는 단계와;

상기 오디오 프로세서를 통해, 상기 중첩 윈도우 변환 행렬 C를 상기 서브프레임 및 상기 서브프레임의 바로 전의 인접한 서브프레임에 적용하는 단계와;

상기 오디오 프로세서를 통해, 상기 블록 변환 행렬 B를 상기 서브프레임에 적용하는 단계를 포함하고,

$$C_{i(j-1)} = B_{ij}^{-1} A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1} A_{ij}$$

$$B_{(i+1)j} = A_{(i+1)(j-1)} B_{ij} A_{i(j-1)}^{-1}$$

상기 직사각형 선형 변환 행렬 A를 분해하는 단계는 관계식 을 푸는 단계를 포함하는 것인,

겹침 변환의 가역적 정수 입력/정수 출력 구현들을 도출하는 방법.

청구항 2

오디오 신호를 인코딩하는 방법으로서,

시간 영역 오디오 샘플들의 시퀀스를 포함하는 오디오 신호의 입력을 수신하는 단계와;

상기 오디오 신호를 적어도 2개의 상이한 서브프레임 크기를 갖는 서브프레임들로 분할하는 단계와;

상기 상이한 크기의 서브프레임들의 가역적 정수 입력 대 정수 출력(reversible integer-input to interger-output)을 이용해 겹침 변환을 수행하는 단계와;

결과적으로 얻은 변환 계수들을 압축된 비트 스트림으로서 인코딩하는 단계를 포함하고,

상기 겹침 변환은 각각의 서브프레임 크기인 상기 서브프레임들에 적용되는 가역적 블록 변환들과, 각각의 인접한 서브프레임들을 중첩시키는 가역적 블록 회전 변환들의 조합으로서 실현되고, 각각의 가역적 블록 회전 변환은 상기 중첩되는 인접한 서브프레임들의 더 작은 쪽의 최대 절반까지 중첩시키며, 서브프레임의 상기 가역적 블록 변환 및 인접한 서브프레임들을 중첩시키는 상기 가역적 블록 회전 변환은 상기 겹침 변환을 분해(factorization)한 것이고,

$$\begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$

상기 겹침 변환은 형태를 갖는 직사각형 선형 변환 행렬 A - 상기 행렬 A 내의 각 서브블록은 상기 서브프레임의 크기가 N인 경우에 N/2 x N/2 행렬이고, i, j는 상기 행렬 서브블록들의 지수들이며, 또한 상기 행렬 A는  $A_{i(j-1)}A_{(i+1)(j-1)}^{-1} = A_{ij}A_{(i+1)j}^{-1}$  라는 속성을 가진 - 의 조합으로서 실현되고,

$$\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$$

상기 직사각형 선형 변환 행렬 A는 형태의 중첩 윈도우 변환 행렬 C와

$$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$$

형태의 블록 변환 행렬 B로 분해되고, 상기 행렬 A, B 및 C는 관계식

$$C_{i(j-1)} = B_{ij}^{-1}A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1}A_{ij}$$

$$B_{(i+1)j} = A_{(i+1)(j-1)}B_{ij}A_{i(j-1)}^{-1} \text{ 을 만족시키는 것인,}$$

인코딩 방법.

**청구항 3**

제2항에 있어서,

상기 겹침 변환은 MLT(modulated lapped transform)인

인코딩 방법.

**청구항 4**

제3항에 있어서,

상기 겹침 변환은 MDCT(modified discrete cosine transform)인

인코딩 방법.

**청구항 5**

제4항에 있어서,

상기 가역적 블록 변환은 블록 기반 DCT-IV형 변환이고, 상기 가역적 블록 회전 변환은 1단 버터플라이형(single stage butterfly type) 변환인

인코딩 방법.

**청구항 6**

오디오 신호를 디코딩하는 방법으로서,

상이한 크기의 서브프레임들로 분할된 오디오 신호의 변환 계수들을 디코딩하는 단계와;

상기 상이한 크기의 서브프레임들의 가역적 정수 입력 대 정수 출력을 이용해 겹침 변환을 수행하는 단계와;

역 변환된 변환 계수들로부터 시간 영역 오디오 신호에 대한 오디오 샘플들을 생성하는 단계와;

상기 오디오 신호를 출력하는 단계를 포함하고,

상기 겹침 변환은 각각의 서브프레임 크기들인 상기 서브프레임들에 적용되는 가역적 블록 변환들과, 각각의 인접한 서브프레임들을 중첩시키는 가역적 블록 회전 변환들의 조합으로서 실현되고, 각각의 가역적 블록 회전 변환은 상기 중첩되는 인접한 서브프레임들의 더 작은 쪽의 최대 절반까지 중첩시키며, 서브프레임의 상기 가역적 블록 변환 및 인접한 서브프레임들을 중첩시키는 상기 가역적 블록 회전 변환은 상기 겹침 변환을 분해한 것이고,

$$\begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$

상기 겹침 변환은 형태를 갖는 직사각형 선형 변환 행렬 A - 상기 행렬 A 내의 각 서브블록은 상기 서브프레임의 크기가 N인 경우에 N/2 x N/2 행렬이고, i, j는 상기 행렬 서브블록들의 지수들이며, 상기 행렬 A는

$$A_{i(j-1)}A_{(i+1)(j-1)}^{-1} = A_{ij}A_{(i+1)j}^{-1}$$

라는 속성을 가진 - 의 조합으로서 실현되고,

$$\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$$

상기 직사각형 선형 변환 행렬 A는 형태의 중첩 윈도우 변환 행렬 C와

$$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$$

형태의 블록 변환 행렬 B로 분해되고, 상기 행렬 A, B 및 C는 관계식

$$C_{i(j-1)} = B_{ij}^{-1}A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1}A_{ij}$$

$$B_{(i+1)j} = A_{(i+1)(j-1)}B_{ij}A_{i(j-1)}^{-1}$$

을 만족시키는 것인,

디코딩 방법.

**청구항 7**

제6항에 있어서,

상기 겹침 변환은 MLT(modulated lapped transform)인

디코딩 방법.

**청구항 8**

제7항에 있어서,

상기 겹침 변환은 MDCT(modified discrete cosine transform)인

디코딩 방법.

**청구항 9**

제8항에 있어서,

상기 가역적 블록 변환은 블록 기반 DCT-IV형 변환이고, 상기 가역적 블록 회전 변환은 1단 버터플라이형

(single stage butterfly type) 변환인

디코딩 방법.

**청구항 10**

복수의 상이한 크기의 서브프레임들로 분할된 입력 오디오 신호를 처리하는 방법에 있어서,

겹침 변환을, 인접한 서브프레임들을 중첩시키는 중첩 영역에 대한 중첩 윈도우 변환들과 상기 서브프레임들에 대한 블록 변환들의 조합들로 분해한 것(factorization)으로 나타내는 단계로서, 상기 분해는

$$\begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$

주어진 서브프레임에 대한 상기 겹침 변환을 행렬을 갖는 직사각형 선형 변환 행렬 A - 상기 행렬 A 내의 각 서브블록은 상기 서브프레임의 크기가 N인 경우에 N/2 x N/2 행렬이고, i, j는 상기 행렬 서브블록들의 지수들이며, 상기 행렬 A는  $A_{i(j-1)}A_{(i+1)(j-1)}^{-1} = A_{ij}A_{(i+1)j}^{-1}$  라는 속성을 가진 - 로서 나타내는 것과;

상기 직사각형 선형 변환 행렬 A를  $\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$  형태의 중첩 윈도우 변환 행렬 C와

$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$  형태의 블록 변환 행렬 B로 분해된 것으로 나타내는 것을 포함하는, 상기 분해한 것으로 나타내는 단계와;

오디오 프로세서에서, 상기 서브프레임을 갖는 오디오 신호의 입력을 수신하는 단계와;

상기 오디오 프로세서를 통해, 상기 중첩 윈도우 변환 행렬 C를 상기 서브프레임 및 상기 서브프레임의 바로 전의 인접한 서브프레임에 적용하는 단계와;

상기 오디오 프로세서를 통해, 상기 블록 변환 행렬 B를 상기 서브프레임에 적용하는 단계를 포함하고,

$$C_{i(j-1)} = B_{ij}^{-1} A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1} A_{ij}$$

상기 행렬 A, B, C는 관계식  $B_{(i+1)j} = A_{(i+1)(j-1)} B_{ij} A_{i(j-1)}^{-1}$  이 성립되는 것인,

입력 오디오 신호 처리 방법.

**청구항 11**

제10항에 있어서,

상기 겹침 변환은 MLT(modulated lapped transform)인

입력 오디오 신호 처리 방법.

**청구항 12**

오디오 신호를 인코딩하는 방법으로서,

시간 영역 오디오 샘플들의 시퀀스를 포함하는 오디오 신호의 입력을 수신하는 단계와;

상기 오디오 신호를 적어도 2개의 상이한 서브프레임 크기를 갖는 서브프레임들로 분할하는 단계와;

상기 상이한 크기의 서브프레임들의 가역적 정수 입력 대 정수 출력(reversible integer-input to interger-

output)을 갖는 겹침 변환을 수행하는 단계와;

결과적으로 얻은 변환 계수들을 압축된 비트 스트림으로서 인코딩하는 단계를 포함하고,

상기 겹침 변환은 중첩 연산자를 적용하고 후속하여 블록 변환 연산자를 적용함으로써 실현되고, 상기 중첩 연산자 및 상기 블록 변환 연산자는 상기 겹침 변환의 분해에 대응하고, 상기 블록 변환 연산자는 중첩될 서브프레임들과 연관된 서브프레임 크기에 기초하고,

상기 중첩 연산자 및 상기 블록 변환 연산자는 각각 중첩 행렬 및 블록 변환 행렬로서 표현될 수 있고,

$$\begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$

주어진 서브프레임에 대한 상기 변환은

형태를 갖는 직사각형

선형 변환 행렬 A로서 표현될 수 있고, 상기 행렬 A 내의 각 서브블록은 상기 서브프레임의 크기가 N인 경우에  $N/2 \times N/2$  행렬이고,  $i, j$ 는 상기 행렬 서브블록들의 지수들이며, 상기 행렬 A는

$$A_{i(j-1)}A_{(i+1)(j-1)}^{-1} = A_{ij}A_{(i+1)j}^{-1}$$

라는 속성을 가지며,

$$\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$$

상기 중첩 연산자는

형태의 중첩 행렬 C로 표현될 수 있고,

$$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$$

상기 블록 변환 연산자는

형태의 행렬 B로 표현될 수 있으며,

$$C_{i(j-1)} = B_{ij}^{-1}A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1}A_{ij}$$

관계식  $B_{(i+1)j} = A_{(i+1)(j-1)}B_{ij}A_{i(j-1)}^{-1}$  이 성립되는

인코딩 방법.

### 청구항 13

오디오 신호를 인코딩하는 방법으로서,

시간 영역 오디오 샘플들의 시퀀스를 포함하는 오디오 신호의 입력을 수신하는 단계와;

상기 오디오 신호를 적어도 2개의 상이한 서브프레임 크기 - 최소의 서브프레임 크기 N은 4보다 큰 짝의 정수임 - 를 갖는 서브프레임들로 분할하는 단계와;

상기 상이한 크기 서브프레임들의 가역적 정수 입력 대 정수 출력(reversible integer-input to interger-output)을 이용해 겹침 변환을 수행하는 단계와;

결과적으로 얻은 변환 계수들을 압축된 비트 스트림으로서 인코딩하는 단계를 포함하고,

상기 겹침 변환은 중첩 연산자를 적용하고 후속하여 블록 변환 연산자를 적용함으로써 실현되고, 상기 중첩 연산자 및 상기 블록 변환 연산자는  $N/2 \times N/2$  중첩 부분행렬(submatrices)의 어레이를 포함하는 중첩 행렬과  $N/2 \times N/2$  변환 부분행렬의 어레이를 포함하는 블록 변환 행렬로 겹침 변환 행렬을 분해한 것에 대응하고, 상기 블록 변환 연산자는 중첩될 서브프레임들과 연관된 서브프레임 크기에 기초하는 것인,

인코딩 방법.

### 청구항 14

제13항에 있어서,  
 상기 겹침 변환은 MLT(modulated lapped transform)인  
 인코딩 방법.

**청구항 15**

제14항에 있어서,  
 상기 겹침 변환은 MDCT(modified discrete cosine transform)인  
 인코딩 방법.

**청구항 16**

제15항에 있어서,  
 상기 블록 변환 연산자는 블록 기반 DCT-IV형 변환에 대응하고, 상기 중첩 연산자는 1단 버터플라이형(single stage butterfly type) 변환에 대응하는  
 인코딩 방법.

**청구항 17**

제13항에 있어서,  
 상기 최소 서브프레임 크기 N은 256, 512, 1024 또는 2048인  
 인코딩 방법.

**청구항 18**

제17항에 있어서,  
 서브프레임에 대한 상기 겹침 변환은 선행 인접 서브프레임과 후행 인접 서브프레임이 상이한 크기를 갖는 경우  
 상기 선행 인접 서브프레임과 상기 후행 인접 서브프레임을 중첩시키기 위해 상이한 크기의 중첩 행렬로 분해되  
 는  
 인코딩 방법.

**청구항 19**

제17항에 있어서,  
 상기 중첩 연산자는 상기 중첩 연산자에 의해 중첩될 2개의 프레임 중 더 작은 프레임에 대한 서브프레임 크기  
 의 절반에 대응하는 중첩을 생성하도록 구성되는  
 인코딩 방법.

**청구항 20**

제13항에 있어서,  
 크기 MN(M은 정수임)의 서브프레임에 대응하는 상기 겹침 변환 행렬의 일부는  $2M \times 2M+2$  부분행렬 - 상기  $2M \times 2M+2$  부분행렬의 원소들은  $N/2 \times N/2$  변환 부분행렬들임 - 로서 표현될 수 있는,  
 인코딩 방법.

**발명의 설명**

**배경 기술**

휴대용 디지털 미디어 플레이어, 음악 저장을 위한 콤팩트 디스크, 및 인터넷을 통한 오디오 배달의 도입으로,  
 이제 디지털 오디오 포맷의 음악 및 그외 다른 오디오 콘텐츠를 저장하고, 구입하며, 배포하는 것이

[0001]

일반적이다. 디지털 오디오 포맷은 사람들이 그들의 개인용 컴퓨터(PC) 또는 휴대용 미디어 플레이어에서 사용할 수 있는 수백 또는 수천 곡의 노래를 갖고 즐길 수 있게 한다.

[0002] **지각 변환 코딩(Perceptual Transform Coding)**

[0003] 오디오의 코딩은 인간 청각의 다양한 지각 모델을 이용하는 코딩 기술을 사용한다. 예를 들어, 강한 음 근처의 많은 약한 음은 마스킹되므로, 코딩될 필요가 없다. 종래의 지각 오디오 코딩에서, 이것은 상이한 주파수 데이터의 적응 양자화로서 이용된다. 지각에 중요한 주파수 데이터는 더 많은 비트, 따라서 더 정교한 양자화가 할당되고, 이와 반대로도 된다.

[0004] 예를 들어, 변환 코딩은 종래에 오디오 신호의 압축을 위한 효율적인 방식으로 알려져 있다. 입력 오디오는 디지털 방식으로 시간 샘플링된다. 변환 코딩에서, 입력 오디오 샘플의 블록은 (예를 들어, 가장 널리 사용되는 MDCT(Modified Discrete Cosine Transform)를 통해) 변환되고, 처리되며, 양자화된다. 변환된 계수의 양자화는 지각 중요도(예를 들어, 인간 청각의 마스킹 효과 및 주파수 감도)에 기초하여, 이를테면 스칼라 양자화기를 통해 실행된다.

[0005] 스칼라 양자화가 사용될 때, 중요도는 상대 가중치에 매핑되고, 각 계수에 대한 양자화기 해상도(스텝 크기)는 그 가중치 및 전체 해상도로부터 구해진다. 전체 해상도는 타깃 품질, 비트 속도 등으로부터 결정될 수 있다. 제공된 스텝 크기에 대해, 각 계수는 0 또는 0이 아닌 정수 값인 수준으로 양자화된다.

[0006] 낮은 비트 속도에서는, 0이 아닌 수준의 계수보다 매우 많은 0 수준의 계수가 있다. 이들은 허프만(Huffman) 코딩과 같은 엔트로피 코딩 방식과 결합될 수 있는 실행 길이(run-length) 코딩을 사용하여 큰 효율로 코딩될 수 있다.

[0007] **중첩 변환 및 가변 윈도우 프레임 크기**

[0008] 많은 오디오 압축 시스템은 오디오 변환 코딩으로 시간-주파수 분석을 실행하기 위해 MLT(Modulated Lapped Transform)(MDCT라고도 알려져 있음)를 이용한다. MLT는 양자화에 의해 재구성된 오디오 신호에 유입된 블록화 아티팩트를 감소시킨다. 더욱 구체적으로, 비중첩 블록이 독립적으로 변환 코딩될 때, 양자화 오류는 디코더에서의 오디오 신호의 재구성시에 블록 경계에서 신호의 불연속성을 생성할 것이다. 오디오의 경우에, 주기적인 클릭 효과가 들린다.

[0009] MLT는 블록을 중첩함으로써 블록화 효과를 감소시킨다. MLT에서, 2개의 연속되는 블록으로부터의 2M 샘플의 "윈도우"는 변조된 코사인 변환이 수행된다. M 변환 계수가 반환된다. 그 다음, 윈도우는 M 샘플만큼 이동되고, 다음의 M 변환 계수 집합이 계산된다. 그러므로, 각각의 윈도우는 이전 윈도우의 마지막 M 샘플과 중첩된다. 중첩은 양자화로 인한 변환 계수의 변경에도 불구하고 재구성된 샘플의 연속성을 향상시킨다.

[0010] 몇몇 오디오 압축 시스템은 오디오의 변화하는 특성에 맞게 조정하기 위해 시간에 따라 윈도우의 크기를 변화시킨다. 오디오 코더는 통상적으로 입력 오디오 신호를 고정 크기의 "프레임들"로 분할하는데, 각 프레임은 코딩의 단위이다(예를 들어, 코딩 테이블 및/또는 파라미터가 각 프레임의 헤더 구역 내에서 보내질 수 있다). 시변 MLT를 사용하는 오디오 압축 시스템에서, 각 프레임은 가변 크기의 하나 이상의 "윈도우"를 포함할 수 있는데, 각 윈도우는 MLT의 단위이다. 일반적으로, 더 큰 윈도우는 코딩 효율에 유리한 반면, 더 작은 크기의 윈도우는 더 양호한 시간 해상도를 제공한다. 따라서, 어디에서 어떤 윈도우 크기를 이용할 것인지를 결정은 인코딩된 신호의 압축 성능 및 청각 품질에 중요하다.

[0011] 오디오 코딩에서의 한 가지 문제는 일반적으로 "프리 에코(pre-echo)"라 칭해진다. 프리 에코는 오디오가 ("과도 현상"이라고 하는) 갑작스런 변화를 겪을 때 발생한다. 변환 코딩에서, 특정 주파수 계수는 일반적으로 양자화된다(즉, 해상도가 감소된다). 변환 계수가 오디오 신호를 재생하기 위해 나중에 역변환될 때, 이 양자화는 시간 영역에서 전체 블록에 걸쳐 퍼지는 양자화 잡음을 끌어들이는다. 본질적으로, 이것은 오히려 코딩 프레임 내의 잡음의 균일한 퍼짐 현상을 야기한다. 일반적으로 프레임의 어떤 부분에 허용되는 잡음은 들릴 수 있어서, 마스킹 수준이 낮은 프레임 부분 동안에 청각적 품질을 손상시킬 수 있다. 실제로, 이 효과는 신호가 낮은 에너지 영역의 바로 다음에 날카로운 공격을 받을 때, 따라서 "프리 에코"일 때 가장 두드러지게 나타난다. 신호가 높은 에너지에서 낮은 에너지로 변화할 때 발생하는 "포스트(post) 에코"는 인간 청각 시스템의 속성으로 인해 지각할 수 있는 청각 품질에 덜 문제된다.

[0012] **중첩된 변환 및 무손실 코딩**

[0013] 중첩 변환은 또한 무손실 코딩에 사용될 수 있다. 다수의 무손실 코딩 기술은 시간 영역에서 오디오 신호 데이터



터에 실행된다. 그러나, 무손실 코딩은 또한 양자화 없이 겹침 변환의 적용 후에 초래된 변환 계수의 엔트로피 코딩 또는 기타 무손실 코딩을 단순히 실행함으로써, 주파수 영역에서 실행될 수 있다. 이러한 주파수 영역 무손실 코딩은 오디오 신호의 손실 및 무손실 압축 버전이 함께 더욱 용이하게 얻어질 수 있게 한다. 그러나, 주파수 영역 무손실 압축은 변환이 가역적이어야 한다. 더욱이, 변환을 가역적이게 하는데 필요한 일관성 있는 계산을 위해, 변환은 정수 구현을 해야 한다.

[0014] 가역적 정수-정수 변환은, 구현이 변환 행렬 내의 직사각형 서브 블록을 사용하는 중첩 변환 설계에 적합하지 않은 1의 결정자를 갖는 정사각형 변환 행렬을 사용하는 구현을 필요로 한다는 점에서 어려움이 있다. 이전에 알려져 있는 가역적 중첩 변환은 또한, 이를테면 위에서 설명된 프리 에코를 감소시키기 위해 가변 서브프레임 크기를 이용하는 오디오 코덱과 호환되지 않는 오디오 신호의 모든 서브프레임에 대해 동일한 서브프레임 구성을 통상적으로 필요로 했다.

[0015] 이전의 가역적 변환 구현에서, MDCT 및 MLT와 같은 변환은  $2N \times N$  변환으로 처리되는데,  $2N \times N$  변환은 가역적으로 된다. 이 절차는 모든 서브프레임이 동일한 크기로 될 때는 알맞지만, 서브프레임 크기가 변할 때는 특별히 잘 맞지는 않는다. 게다가, 중첩된 직교 변환은 이해하고 수정하기 어려운 구현을 갖는다.

**발명의 내용**

[0016] <발명의 요약>

[0017] 다음의 상세한 설명은 가변 서브프레임 크기를 갖는 중첩 변환을 사용하여 변환 영역에서의 무손실 오디오 압축을 위한 다양한 오디오 인코딩/디코딩 기술 및 도구에 관한 것이다. 이 기술은 MLT(modulated lapped transform)와 같은 변환을 2개의 이동된 블록 변환(하나는 중첩 변환이고, 다른 하나는 (임의의 유형의 표준 DCT와 같은) 블록 변환임)의 조합으로 분해함으로써 형성되는, 가변 프레임 크기에 대한 가역적 정수-정수 중첩 변환의 구현을 사용한다.

[0018] 이 요약은 아래의 상세한 설명에서 더욱 설명되는 개념들의 선택된 개념을 단순화된 형태로 소개하기 위해 제공된다. 이 요약은 청구된 주제의 핵심적인 특징이나 중요한 특징을 확인하고자 하는 것도 아니고, 청구된 주제의 범위를 한정하는데 보조적으로 사용하고자 하는 것도 아니다. 본 발명의 추가 특징 및 장점은 첨부 도면을 참조하여 진행되는 실시예의 다음의 상세한 설명으로부터 명백해질 것이다.

**도면의 간단한 설명**

- [0019] 도 1은 다양한 설명된 실시예가 구현될 수 있는 일반화된 운영 환경의 블록도.
- 도 2, 3, 4 및 5는 다양한 설명된 실시예가 구현될 수 있는 일반화된 인코더 및/또는 디코더의 블록도.
- 도 6은 오디오 비트 스트림을 위한 예시적인 서브프레임 구성을 도시한 도면.
- 도 7은 겹침 변환의 중첩 연산자 부분이 적용되는 도 6의 예시적인 서브프레임 구성의 중첩 영역을 도시한 도면.
- 도 8은 도 6의 예시적인 서브프레임 구성에서 겹침 변환의 블록 변환 부분이 적용되는 블록을 도시한 도면.

**발명을 실시하기 위한 구체적인 내용**

[0020] 오디오 정보를 표시하고, 코딩하며, 디코딩하는 다양한 기술 및 도구가 설명된다. 이들 기술 및 도구는 매우 낮은 비트 속도에서도 고품질 오디오 콘텐츠의 생성, 배포 및 재생을 용이하게 한다.

[0021] 여기에서 설명된 다양한 기술 및 도구는 독립적으로 사용될 수 있다. 일부 기술 및 도구는 (예를 들어, 결합된 인코딩 및/또는 디코딩 프로세스의 서로 다른 단계에서) 조합하여 사용될 수 있다.

[0022] 다양한 기술은 처리 동작의 순서도를 참조하여 아래에 설명된다. 순서도에 도시된 다양한 처리 동작은 더 적은 수의 동작으로 통합되거나, 더 많은 수의 동작으로 분리될 수 있다. 단순하게 하기 위해, 특정 순서도에 도시된 동작과 그의 다른 곳에서 설명된 동작의 관계는 종종 표시되지 않는다. 많은 경우에, 순서도의 동작은 순서가 변경될 수 있다.

[0023] 상세한 설명의 대부분은 오디오 정보의 표시, 코딩 및 디코딩에 관해 설명한다. 오디오 정보의 표시, 코딩 및 디코딩을 위해 여기에 설명된 다수의 기술 및 도구는 또한 비디오 정보, 정지 이미지 정보, 또는 단일 또는 다

중 채널에서 보내진 기타 미디어 정보에 적용될 수 있다.

**[0024] I. 컴퓨팅 환경**

**[0025]** 도 1은 설명된 실시예가 구현될 수 있는 적합한 컴퓨팅 환경(100)의 일반화된 예를 도시한 것이다. 컴퓨팅 환경(100)은 설명된 실시예가 다양한 범용 또는 특수 목적 컴퓨팅 환경에서 구현될 수 있기 때문에, 용도 또는 기능의 범위에 관해 어떤 제한을 암시하고자 하는 것이 아니다.

**[0026]** 도 1과 관련하여, 컴퓨팅 환경(100)은 적어도 하나의 처리 장치(110) 및 메모리(120)를 포함한다. 도 1에서, 이 가장 기본적인 구성(130)은 점선 내에 포함된다. 처리 장치(110)는 컴퓨터 실행가능 명령어를 실행하고, 실제 또는 가상 프로세서일 수 있다. 다중 처리 시스템에서, 다중 처리 장치는 처리 능력을 증가시키기 위한 컴퓨터 실행가능 명령어를 실행한다. 처리 장치는 또한 중앙 처리 장치 및 보조 프로세서, 및/또는 전용 또는 특수 목적 처리 장치(예를 들어, 오디오 프로세서)를 포함할 수 있다. 메모리(120)는 휘발성 메모리(예를 들어, 레지스터, 캐시, RAM), 비휘발성 메모리(예를 들어, ROM, EEPROM, 플래시 메모리), 또는 이 두 개의 어떤 조합일 수 있다. 메모리(120)는 하나 이상의 설명된 실시예에 따른 하나 이상의 오디오 처리 기술 및/또는 시스템을 구현하는 소프트웨어(180)를 저장한다.

**[0027]** 컴퓨팅 환경은 추가적인 특징을 가질 수 있다. 예를 들어, 컴퓨팅 환경(100)은 저장소(140), 하나 이상의 입력 장치(150), 하나 이상의 출력 장치(160) 및 하나 이상의 통신 연결(170)을 포함한다. 버스, 컨트롤러 또는 네트워크와 같은 상호접속 메커니즘(도시 생략)은 컴퓨팅 환경(100)의 구성요소들을 상호접속시킨다. 통상적으로, 운영 체제 소프트웨어(도시 생략)는 컴퓨팅 환경(100)에서 실행되는 소프트웨어를 위한 운영 환경을 제공하고, 컴퓨팅 환경(100)의 구성요소의 활동을 조정한다.

**[0028]** 저장소(140)는 이동식 또는 비이동식일 수 있고, 자기 디스크, 자기 테이프 또는 카세트, CD, DVD, 또는 정보를 저장하기 위해 사용될 수 있고 컴퓨팅 환경(100) 내에서 액세스될 수 있는 임의의 다른 매체를 포함한다. 저장소(140)는 소프트웨어(180)를 위한 명령어를 저장한다.

**[0029]** 입력 장치(들)(150)는 터치 입력 장치, 이클레멘 키보드, 마우스, 펜, 터치스크린 또는 트랙볼, 음성 입력 장치, 스캐닝 장치, 또는 컴퓨팅 환경(100)에 입력을 제공하는 다른 장치일 수 있다. 오디오 또는 비디오의 경우에, 입력 장치(들)(150)는 마이크, 사운드 카드, 비디오 카드, TV 튜너 카드, 또는 아날로그 또는 디지털 형태의 오디오 또는 비디오 입력을 받아들이는 유사한 장치, 또는 오디오 또는 비디오 샘플을 컴퓨팅 환경 내로 읽어들이는 CD 또는 DVD일 수 있다. 출력 장치(들)(160)는 디스플레이, 프린터, 스피커, CD/DVD 라이터(writer), 네트워크 어댑터, 또는 컴퓨팅 환경(100)으로부터 출력을 제공하는 다른 장치일 수 있다.

**[0030]** 통신 연결(들)(170)은 통신 매체를 통해 하나 이상의 다른 컴퓨팅 엔티티에 통신할 수 있게 한다. 통신 매체는 컴퓨터 실행가능 명령어, 오디오 또는 비디오 정보, 또는 디지털 신호의 기타 데이터와 같은 정보를 전달한다. 피변조 데이터 신호는 신호 내에 정보를 인코딩하는 방식으로 신호의 하나 이상의 특성이 설정되거나 변경된 신호이다. 예시적이고 비제한적으로, 통신 매체는 전기, 광, RF, 적외선, 음향 또는 기타 반송파로 구현된 유선 또는 무선 기술을 포함한다.

**[0031]** 실시예는 일반적으로 컴퓨터 판독가능 매체와 관련하여 설명될 수 있다. 컴퓨터 판독가능 매체는 컴퓨팅 환경 내에서 액세스될 수 있는 임의의 사용 가능한 매체이다. 예시적이고 비제한적으로, 컴퓨팅 환경(100)에서, 컴퓨터 판독가능 매체는 메모리(120), 저장소(140) 및 위의 임의의 조합을 포함한다.

**[0032]** 실시예는 일반적으로 컴퓨터 실행가능 명령어, 이클레멘 실제 또는 가상의 대상 프로세서 상의 컴퓨팅 환경에서 실행되는 프로그램 모듈에 포함된 것들과 관련하여 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 작업을 수행하거나 특정 데이터 유형을 구현하는 루틴, 프로그램, 라이브러리, 개체, 클래스, 구성요소, 데이터 구조 등을 포함한다. 프로그램 모듈의 기능은 다양한 실시예에서 원하는 대로 프로그램 모듈 사이에서 결합되거나 분할될 수 있다. 프로그램 모듈을 위한 컴퓨터 실행가능 명령어는 로컬 또는 분산 컴퓨팅 환경에서 실행될 수 있다.

**[0033]** 설명을 위해, 상세한 설명에서는 컴퓨팅 환경에서의 컴퓨터 작업을 설명하기 위해 "결정하다", "수신하다" 및 "수행하다"와 같은 용어를 사용한다. 이들 용어는 컴퓨터에 의해 수행된 작업에 대한 고급 추상 개념이고, 인간에 의해 수행된 동작과 혼동되어서는 안 된다. 이들 용어에 대응하는 실제 컴퓨터 작업은 구현에 따라 다르다.

**[0034] II. 예시적인 인코더 및 디코더**

- [0035] 도 2는 하나 이상의 설명된 실시예가 구현될 수 있는 제1 오디오 인코더(200)를 도시한 것이다. 인코더(200)는 변환 기반의 지각 오디오 인코더(200)이다. 도 3은 대응하는 오디오 디코더(300)를 도시한 것이다.
- [0036] 도 4는 하나 이상의 설명된 실시예가 구현될 수 있는 제2 오디오 인코더(400)를 도시한 것이다. 인코더(400)는 또한 변환 기반의 지각 오디오 인코더이지만, 인코더(400)는 다중 채널 오디오를 처리하는 모듈과 같은 추가 모듈을 포함한다. 도 5는 대응하는 오디오 디코더(500)를 도시한 것이다.
- [0037] 도 2 내지 5에 도시된 시스템이 일반화된 것이긴 하지만, 각각은 현실 세계 시스템에서 찾아볼 수 있는 특성이 있다. 여하튼, 인코더 및 디코더 내의 모듈들 사이에 도시된 관계는 인코더 및 디코더 내의 정보의 흐름을 나타내고; 그의 다른 관계는 단순화를 위해 나타내지 않는다. 원하는 압축의 구현 및 유형에 의존하여, 인코더 또는 디코더의 모듈은 추가되고, 생략되며, 다수의 모듈로 분할되고, 그의 다른 모듈과 결합되거나, 및/또는 유사한 모듈로 교체될 수 있다. 대안적인 실시예에서, 상이한 모듈 및/또는 기타 구성을 갖는 인코더 또는 디코더는 오디오 데이터, 또는 하나 이상의 설명된 실시예에 따른 어떤 다른 유형의 데이터를 처리한다.
- [0038] **A. 제1 오디오 인코더**
- [0039] 인코더(200)는 소정의 샘플링 깊이 및 속도에서 시계열의 입력 오디오 샘플(205)을 수신한다. 입력 오디오 샘플(205)은 다중 채널 오디오(예를 들어, 스테레오) 또는 모노 오디오를 위한 것이다. 인코더(200)는 오디오 샘플(205)을 압축하고, 인코더(200)의 다양한 모듈에 의해 생성된 정보를 멀티플렉싱하여, WMA 포맷과 같은 압축 포맷, ASF(Advanced Streaming Format)와 같은 컨테이너 포맷, 또는 그의 다른 압축 또는 컨테이너 포맷으로 비트 스트림(295)을 출력한다.
- [0040] 주파수 변환기(210)는 오디오 샘플(205)을 수신하여, 주파수(또는 스펙트럼) 영역의 데이터로 변환한다. 예를 들어, 주파수 변환기(210)는 프레임의 오디오 샘플(205)을 서브프레임 블록으로 분할하는데, 이들 서브프레임 블록은 가변 시간 해상도를 허용하도록 가변 크기를 갖는다. 블록은 나중의 양자화에 의해 그 밖에 도입될 수 있는 블록들 사이의 지각 불연속성을 감소시키기 위해 중첩될 수 있다. 주파수 변환기(210)는 시변 MLT, 변조된 DCT(MDCT), 소정의 다른 종류의 MLT 또는 DCT, 또는 소정의 다른 유형의 변조 또는 비변조, 중첩 또는 비중첩 주파수 변환을 블록에 적용하거나, 또는 부대역(sub-band) 또는 웨이브릿(wavelet) 코딩을 사용한다. 주파수 변환기(210)는 스펙트럼 계수 데이터의 블록을 출력하고, 블록 크기와 같은 사이드 정보를 멀티플렉서(MUX)(280)에 출력한다.
- [0041] 다중 채널 오디오 데이터의 경우에, 다중 채널 변환기(220)는 다수의 원래의 독립적으로 코딩된 채널을 함께 코딩된 채널로 변환할 수 있다. 또는, 다중 채널 변환기(220)는 좌우 채널을 독립적으로 코딩된 채널로서 통과시킬 수 있다. 다중 채널 변환기(220)는 사용된 채널 모드를 나타내는 사이드 정보를 생성하여 MUX(280)에 보낸다. 인코더(200)는 다중 채널 변환 후 오디오 데이터의 블록에 다중 채널 재행렬화(rematrixing)를 적용할 수 있다.
- [0042] 지각 모델러(230)는 제공된 비트 속도에 대한 재구성된 오디오 신호의 인식 품질을 개선하기 위해 인간 청각 시스템의 속성을 모델링한다. 지각 모델러(230)는 다양한 청각 모델 중의 임의의 것을 사용하고, 여기(excitation) 패턴 정보 또는 그의 다른 정보를 가중화기(weighter)(240)에 전달한다. 예를 들어, 청각 모델은 통상적으로 인간 청력의 범위 및 임계 대역(예를 들어, 바크(Bark) 대역)을 고려한다. 범위 및 임계 대역 외에, 오디오 신호들 사이의 상호작용이 극적으로 지각에 영향을 미칠 수 있다. 게다가, 청각 모델은 소리의 인간 지각의 신체적 또는 신경적 측면에 관한 다양한 기타 요인을 고려할 수 있다.
- [0043] 지각 모델러(230)는 잡음의 가청도를 감소시키도록 오디오 데이터 내의 잡음을 성형하기 위해 가중화기(240)가 사용하는 정보를 출력한다. 예를 들어, 다양한 기술 중의 임의의 기술을 사용하여, 가중화기(240)는 수신된 정보에 기초하여 (때때로 마스크라 불리는) 양자화 행렬에 대한 가중 계수를 생성한다. 양자화 행렬에 대한 가중 계수는 행렬 내의 다수의 양자화 대역의 각각에 대한 가중치를 포함하는데, 양자화 대역은 주파수 계수의 주파수 범위이다. 그러므로, 가중 계수는 잡음/양자화 오류가 양자화 대역을 가로질러 퍼지는 비율을 나타냄으로써, 덜 들리는 대역에 잡음을 더 많이 삽입하고, 이와 반대로 하여, 잡음의 가청도를 최소화할 목적으로, 잡음/양자화 오류의 스펙트럼/시간 분포를 제어한다.
- [0044] 그 다음, 가중화기(240)는 다중 채널 변환기(220)로부터 수신된 데이터에 가중 계수를 적용한다.
- [0045] 양자화기(250)는 가중화기(240)의 출력을 양자화하고, 양자화된 계수 데이터를 생성하여 엔트로피 인코더(260)에 보내고, 양자화 스텝 크기를 포함하는 사이드 정보를 MUX(280)에 보낸다. 도 2에서, 양자화기(250)는 적용

적인 균일한 스칼라 양자화기이다. 양자화기(250)는 동일한 양자화 스텝 크기를 각 스펙트럼 계수에 적용하지만, 양자화 스텝 크기 자체는 엔트로피 인코더(260) 출력의 비트 속도에 영향을 미치기 위해 하나의 반복하는 양자화 루프에서 다음 루프로 변경될 수 있다. 그의 다른 종류의 양자화는 불균일 벡터 양자화, 및/또는 비적응 양자화이다.

[0046] 엔트로피 인코더(260)는 예를 들어, 실행 수준 코딩 및 벡터 가변 길이 코딩을 실행하여, 양자화기(250)로부터 수신된 양자화된 계수 데이터를 무손실로 압축한다. 엔트로피 인코더(260)는 오디오 정보의 인코딩에 소비된 비트의 수를 계산하여, 이 정보를 속도/품질 컨트롤러(270)에 전달할 수 있다.

[0047] 컨트롤러(270)는 인코더(200) 출력의 비트 속도 및/또는 품질을 조정하기 위해 양자화기(250)와 함께 작업한다. 컨트롤러(270)는 비트 속도 및 품질 제약조건을 충족시킬 목적으로 양자화 스텝 크기를 양자화기(250)에 출력한다.

[0048] 게다가, 인코더(200)는 잡음 대체 및/또는 대역 잘라내기를 오디오 데이터의 블록에 적용할 수 있다.

[0049] MUX(280)는 엔트로피 인코더(260)로부터 수신된 엔트로피 인코딩 데이터와 함께 오디오 인코더(200)의 그의 다른 모듈로부터 수신된 사이드 정보를 멀티플렉싱한다. MUX(280)는 인코더(200)에 의해 출력될 비트 스트림(295)을 저장하는 가상 버퍼를 포함할 수 있다.

[0050] **B. 제1 오디오 디코더**

[0051] 디코더(300)는 사이드 정보뿐만 아니라 엔트로피 인코딩 데이터를 포함하는 압축된 오디오 정보의 비트 스트림(305)을 수신하는데, 디코더(300)는 사이드 정보로부터 오디오 샘플(395)을 재구성한다.

[0052] 디멀티플렉서(DEMUX)(310)는 비트 스트림(305) 내의 정보를 구문 분석하여, 이 정보를 디코더(300)의 모듈에 보낸다. DEMUX(310)는 오디오 복잡도의 변동, 네트워크 지터 및/또는 기타 요인으로 인한 비트 속도의 단기간 변화를 보상하기 위해 하나 이상의 버퍼를 포함한다.

[0053] 엔트로피 디코더(320)는 DEMUX(310)로부터 수신된 엔트로피 코드를 무손실로 압축 해제하여, 양자화된 스펙트럼 계수 데이터를 생성한다. 엔트로피 디코더(320)는 통상적으로 인코더에서 사용된 엔트로피 인코딩 기술과 반대로 적용한다.

[0054] 역 양자화기(330)는 DEMUX(310)로부터 양자화 스텝 크기를 수신하고, 엔트로피 디코더(320)로부터 양자화된 스펙트럼 계수 데이터를 수신한다. 역 양자화기(330)는 주파수 계수 데이터를 부분적으로 재구성하거나 그 밖에 역 양자화를 실행하기 위해 양자화 스텝 크기를 양자화된 주파수 계수 데이터에 적용한다.

[0055] DEMUX(310)로부터, 잡음 생성기(340)는 잡음의 형태에 대한 임의의 파라미터뿐만 아니라 데이터 블록 내의 어느 대역이 잡음 대체 대역인지 나타내는 정보를 수신한다. 잡음 생성기(340)는 표시된 대역에 대한 패턴을 생성하고, 이 정보를 역 가중화기(350)에 전달한다.

[0056] 역 가중화기(350)는 DEMUX(310)로부터 가중 계수를 수신하고, 잡음 생성기(340)로부터 임의의 잡음 대체 대역에 대한 패턴을 수신하며, 역 양자화기(330)로부터 부분적으로 재구성된 주파수 계수 데이터를 수신한다. 필요에 따라, 역 가중화기(350)는 가중 계수를 압축 해제한다. 역 가중화기(350)는 잡음이 대체되지 않은 대역에 대한 부분적으로 재구성된 주파수 계수 데이터에 가중 계수를 적용한다. 그 다음, 역 가중화기(350)는 잡음 생성기(340)로부터 수신된 잡음 대체 대역에 대한 잡음 패턴을 추가해 넣는다.

[0057] 역 다중 채널 변환기(360)는 재구성된 스펙트럼 계수 데이터를 역 가중화기(350)로부터 수신하고, 채널 모드 정보를 DEMUX(310)로부터 수신한다. 다중 채널 오디오가 독립적으로 코딩된 채널 내에 있으면, 역 다중 채널 변환기(360)는 채널을 통과시킨다. 다중 채널 데이터가 함께 코딩된 채널 내에 있으면, 역 다중 채널 변환기(360)는 데이터를 독립적으로 코딩된 채널로 변환한다.

[0058] 역 주파수 변환기(370)는 DEMUX(310)로부터의 블록 크기와 같은 사이드 정보뿐만 아니라 다중 채널 변환기(360)에 의해 출력된 스펙트럼 계수 데이터를 수신한다. 역 주파수 변환기(370)는 인코더에서 사용된 주파수의 변환과 반대로 적용하고, 재구성된 오디오 샘플(395)의 블록을 출력한다.

[0059] **C. 제2 오디오 인코더**

[0060] 도 4와 관련하여, 인코더(400)는 소정의 샘플링 깊이 및 속도에서 시계열의 입력 오디오 샘플(405)을 수신한다. 입력 오디오 샘플(405)은 다중 채널 오디오(예를 들어, 스테레오, 서라운드) 또는 모노 오디오를 위한 것이다.

인코더(400)는 오디오 샘플(405)을 압축하고, 인코더(400)의 다양한 모듈에 의해 생성된 정보를 멀티플렉싱하여, WMA Pro 포맷과 같은 압축 포맷, ASF와 같은 컨테이너 포맷, 또는 그외 다른 압축 또는 컨테이너 포맷으로 비트 스트림(495)을 출력한다.

- [0061] 인코더(400)는 오디오 샘플(405)을 위한 다수의 인코딩 모드들 사이에서 선택한다. 도 4에서, 인코더(400)는 혼합/순수 무손실 코딩 모드와 손실 코딩 모드 사이에서 전환한다. 무손실 코딩 모드는 혼합/순수 무손실 코딩(472)를 포함하고, 통상적으로 고품질(및 높은 비트 속도) 압축을 위해 사용된다. 손실 코딩 모드는 가중화기(442) 및 양자화기(460)와 같은 구성요소를 포함하고, 통상적으로 조정 가능한 품질(및 제어된 비트 속도) 압축을 위해 사용된다. 선택 결정은 사용자 입력 또는 그외 다른 기준에 따라 좌우된다.
- [0062] 다중 채널 오디오 데이터의 손실 코딩의 경우에, 다중 채널 전처리기(pre-processor)(410)는 시간 영역 오디오 샘플(405)을 선택적으로 다시 행렬화한다. 예를 들어, 다중 채널 전처리기(410)는 오디오 샘플(405)을 선택적으로 다시 행렬화하여, 하나 이상의 코딩된 채널을 선택하지 않거나, 인코더(400)에서 채널간 상관관계를 증가시키거나 하지만, 디코더(500)에서 (소정 형태로) 재구성을 할 수 있게 한다. 다중 채널 전처리기(410)는 다중 채널 전처리기를 위한 명령어와 같은 사이드 정보를 MUX(490)에 보낼 수 있다.
- [0063] 윈도 모듈(420)은 오디오 입력 샘플(405)의 프레임을 서브프레임 블록(윈도)으로 분할한다. 윈도는 시변 크기 및 윈도 형성 기능을 가질 수 있다. 인코더(400)가 손실 코딩을 사용할 때, 가변 크기 윈도는 가변 시간 해상도를 허용한다. 윈도 모듈(420)은 분할된 데이터의 블록을 출력하고, 블록 크기와 같은 사이드 정보를 MUX(490)에 출력한다.
- [0064] 도 4에서, 타일 구성기(tile configurer)(422)는 다중 채널 오디오의 프레임을 채널마다 분할한다. 타일 구성기(422)는 품질/비트 속도가 허용하는 경우에, 프레임 내의 각 채널을 독립적으로 분할한다. 이것은 예를 들어, 타일 구성기(422)가, 더 작은 윈도를 갖는 특정 채널에서 나타나는 과도현상을 분리시킬 수 있게 하지만, 그외 다른 채널에서 주파수 해상도 또는 압축 효율을 위해 더 큰 윈도를 사용할 수 있게 한다. 이것은 채널마다 과도현상을 분리시킴으로써 압축 효율을 개선할 수 있지만, 개별 채널에서의 분할을 지정하는 추가 정보가 많은 경우에 요구된다. 시간에 맞춰 같은 곳에 배치되는 동일한 크기의 윈도는 다중 채널 변환을 통해 추가 중복을 감소시킬 자격이 주어질 수 있다. 그러므로, 타일 구성기(422)는 시간에 맞춰 같은 곳에 배치되는 동일한 크기의 윈도를 타일로서 그룹화한다.
- [0065] 주파수 변환기(430)는 오디오 샘플을 수신하고, 도 2의 주파수 변환기(210)에 대해 위에서 설명된 것과 같은 변환을 적용하여, 오디오 샘플을 주파수 영역의 데이터로 변환한다. 주파수 변환기(430)는 스펙트럼 계수 데이터의 블록을 가중화기(442)에 출력하고, 블록 크기와 같은 사이드 정보를 MUX(490)에 출력한다. 주파수 변환기(430)는 주파수 계수와 사이드 정보 둘 다를 지각 모델러(440)에 출력한다.
- [0066] 지각 모델러(440)는 일반적으로 도 2의 지각 모델러(230)와 관련하여 위에서 설명된 바와 같이, 청각 모델에 따라 오디오 데이터를 처리하여, 인간 청각 시스템의 속성을 모델링한다.
- [0067] 가중화기(442)는 일반적으로 도 2의 가중화기(240)와 관련하여 위에서 설명된 바와 같이, 지각 모델러(440)로부터 수신된 정보에 기초하여 양자화 행렬을 위한 가중 계수를 생성한다. 가중화기(442)는 가중 계수를 주파수 변환기(430)로부터 수신된 데이터에 적용한다. 가중화기(442)는 양자화 행렬 및 채널 가중 계수와 같은 사이드 정보를 MUX(490)에 출력한다. 양자화 행렬은 압축될 수 있다.
- [0068] 다중 채널 오디오 데이터의 경우에, 다중 채널 변환기(450)는 채널간 상관 관계를 이용하기 위해 다중 채널 변환을 적용할 수 있다. 예를 들어, 다중 채널 변환기(450)는 타일 내의 채널 및/또는 양자화 대역의 전부가 아니라 일부에 다중 채널 변환을 선택적으로 유연성 있게 적용한다. 다중 채널 변환기(450)는 미리 정의된 행렬 또는 사용자 지정 행렬을 선택적으로 사용하고, 효율적인 압축을 사용자 지정 행렬에 적용한다. 다중 채널 변환기(450)는 예를 들어, 다중 채널 변환이 사용되었고 타일의 일부를 다중 채널 변환시켰다는 것을 나타내는 사이드 정보를 생성하여 MUX(490)에 보낸다.
- [0069] 양자화기(460)는 다중 채널 변환기(450)의 출력을 양자화하고, 양자화된 계수 데이터를 생성하여 엔트로피 인코더(470)에 보내고, 양자화 스텝 크기를 포함하는 사이드 정보를 생성하여 MUX(490)에 보낸다. 도 4에서, 양자화기(460)는 타일 당 양자화 계수를 계산하는 적응적인 균일한 스칼라 양자화기이지만, 양자화기(460)는 대신에 소정의 다른 종류의 양자화를 실행할 수 있다.
- [0070] 엔트로피 인코더(470)는 일반적으로 도 2의 엔트로피 인코더(260)와 관련하여 위에서 설명된 바와 같이, 양자화

기(460)로부터 수신된 양자화된 계수 데이터를 무손실 압축한다.

- [0071] 컨트롤러(480)는 인코더(400) 출력의 비트 속도 및/또는 품질을 조정하기 위해 양자화기(460)와 함께 작업한다. 컨트롤러(480)는 품질 및/또는 비트 속도 제약조건을 충족시킬 목적으로 양자화 계수를 양자화기(460)에 출력한다.
- [0072] 혼합/순수 무손실 인코더(472) 및 관련된 엔트로피 인코더(474)는 혼합/순수 무손실 코딩 모드 동안 오디오 데이터를 압축한다. 인코더(400)는 전체 시퀀스에 대해 혼합/순수 무손실 코딩 모드를 사용하거나, 또는 프레임마다, 블록마다, 타일마다 또는 그외 다른 방식으로 코딩 모드들 사이에서 전환한다.
- [0073] MUX(490)는 엔트로피 인코더(470, 474)로부터 수신된 엔트로피 인코딩 데이터와 함께 오디오 인코더(400)의 그외 다른 모듈로부터 수신된 사이트 정보를 멀티플렉싱한다. MUX(490)는 속도 제어 또는 그외 다른 목적을 위해 하나 이상의 버퍼를 포함한다.
- [0074] **D. 제2 오디오 디코더**
- [0075] 도 5를 참조하면, 제2 오디오 디코더(500)는 압축된 오디오 정보의 비트 스트림(505)을 수신한다. 비트 스트림(505)은 사이트 정보뿐만 아니라 엔트로피 인코딩 데이터를 포함하는데, 이 정보로부터 디코더(500)는 오디오 샘플(595)을 재구성한다.
- [0076] DEMUX(510)는 비트 스트림(505) 내의 정보를 구문 분석하여, 이 정보를 디코더(500)의 모듈에 보낸다. DEMUX(510)는 오디오 복잡도의 변동, 네트워크 지터 및/또는 기타 요인으로 인한 비트 속도의 단기간 변화를 보상하기 위해 하나 이상의 버퍼를 포함한다.
- [0077] 엔트로피 디코더(520)는 통상적으로 인코더(400)에서 사용된 엔트로피 인코딩 기술과 반대로 적용하여, DEMUX(510)로부터 수신된 엔트로피 코드를 무손실로 압축 해제한다. 디코딩 데이터가 손실 코딩 모드로 압축되었을 때, 엔트로피 디코더(520)는 양자화된 스펙트럼 계수 데이터를 생성한다.
- [0078] 혼합/순수 무손실 디코더(522) 및 관련된 엔트로피 디코더(들)(520)는 혼합/순수 무손실 코딩 모드 동안 무손실 인코딩된 오디오 데이터를 압축 해제한다.
- [0079] 타일 구성 디코더(530)는 DEMUX(590)로부터 프레임에 대한 타일의 패턴을 나타내는 정보를 수신하고, 필요한 경우에 이 정보를 디코딩한다. 타일 패턴 정보는 엔트로피 인코딩되거나, 그 밖에 다른 파라미터로 표시될 수 있다. 그 다음, 타일 구성 디코더(530)는 타일 패턴 정보를 디코더(500)의 다양한 다른 모듈에 보낸다.
- [0080] 역 다중 채널 변환기(540)는 타일 패턴 정보를 타일 구성 디코더(530)로부터 수신하고, 예를 들어, 다중 채널 변환이 사용되었고 타일의 일부를 변환시켰다는 것을 나타내는 사이트 정보를 DEMUX(510)로부터 수신할 뿐만 아니라, 양자화된 스펙트럼 계수 데이터를 엔트로피 디코더(520)로부터 수신한다. 이 정보를 사용하여, 역 다중 채널 변환기(540)는 필요에 따라 변환 행렬을 압축 해제하고, 하나 이상의 역 다중 채널 변환을 오디오 데이터에 선택적으로 유연성 있게 적용한다.
- [0081] 역 양자화기/가중화기(550)는 DEMUX(510)로부터 양자화 행렬뿐만 아니라 타일 및 채널 양자화 계수와 같은 정보를 수신하고, 역 다중 채널 변환기(540)로부터 양자화된 스펙트럼 계수 데이터를 수신한다. 역 양자화기/가중화기(550)는 수신된 가중 계수 정보를 필요에 따라 압축 해제한다. 그 다음, 양자화기/가중화기(550)는 역 양자화 및 가중화를 실행한다.
- [0082] 역 주파수 변환기(560)는 DEMUX(510)로부터 사이트 정보를 수신하고, 타일 구성 디코더(530)로부터 타일 패턴 정보를 수신할 뿐만 아니라, 역 양자화기/가중화기(550)에 의해 출력된 스펙트럼 계수 데이터를 수신한다. 역 주파수 변환기(570)는 인코더에서 사용된 주파수 변환과 반대로 적용하고, 중첩기/가산기(overlapper/adder)(570)에 블록을 출력한다.
- [0083] 타일 구성 디코더(530)로부터 타일 패턴 정보를 수신하는 것 이외에, 중첩기/가산기(570)는 디코딩된 정보를 역 주파수 변환기(560) 및/또는 혼합/순수 무손실 디코더(522)로부터 수신한다. 중첩기/가산기(570)는 오디오 데이터를 필요에 따라 중첩하고 가산하며, 서로 다른 모드로 인코딩된 오디오 데이터의 프레임 또는 그외 다른 시퀀스를 인터리빙한다.
- [0084] 다중 채널 후처리(580)는 중첩기/가산기(570)에 의해 출력된 시간 영역 오디오 샘플을 선택적으로 다시 행렬화한다. 비트 스트림-제어 후처리를 위해, 후처리 변환 행렬은 시간에 따라 변하고, 신호로 보내지거나 비트 스트림(505)에 포함된다.

[0085] **III. 가변 서브프레임 크기에 관한 가역적 MLT(Modulated Lapped Transform)**

[0086] 위에서 설명된 바와 같이, 대표적인 인코더(예를 들어, 타일 구성기(422))는 이를테면 프리 에코 효과를 방지하기 위한 오디오 내의 과도현상을 위한 더 작은 서브프레임 크기와 코딩 효율을 위한 더 큰 서브프레임 크기의 사용 사이에서 균형을 맞추기 위해, 가변 서브프레임 크기를 사용하여 오디오 신호를 인코딩한다. 인코더는 서브프레임 크기를 적응 가능하게 선택하기 위해 다양한 기술을 이용할 수 있는데, 이를 테면 Chen 등 저의 "ADAPTIVE WINDOW-SIZE SELECTION IN TRANSFORM CODING"이라는 U.S. 특허 출원 공개 No. US-2003-0115052-A1에 설명된 것들을 이용할 수 있다.

[0087] 도 6은 대표적인 인코더에 의해 생성된 예시적인 서브프레임 구성을 도시한 것으로, 오디오 샘플의 한 구역은 256, 512, 512, 1024 및 512 샘플 크기를 갖는 서브프레임(611-615)으로 분할된다. 도 6에 도시된 서브프레임 크기는 단지 예시적인 인코더 구현에서 사용 가능한 서브프레임 크기 선택사항을 나타낸 것일 뿐이고, 예시를 위해 선택된 것이다. 다양한 대안적인 인코더 구현에서, 인코더는 그외 다른 서브프레임 크기, 및/또는 서브프레임 크기의 더 많거나 적은 선택사항을 사용할 수 있다.

[0088] 또한 위에서 설명된 바와 같이, 대표적인 인코더는 (예를 들어, 주파수 변환기(430)에 의해) 오디오 샘플에 주파수 변환을 적용한다. 대표적인 인코더에서 사용된 주파수 변환은 MDCT(Modified Discrete Cosine Transform)와 같은 일종의 MLT(Modulated Lapped Transform)이다. 이 MLT는 각 서브프레임에 대해 이전 및 이후 서브프레임 내의 샘플과 중첩된다. 달리 말하면, 변환은 현재의 서브프레임 내의 샘플 이외에, 이전 및 이후 서브프레임 내의 샘플의 중첩 부분으로부터 입력을 얻는다. 이 예에서, 더 작은 서브프레임 크기(즉, 인접한 것과 현재의 서브프레임 크기 중의 더 작은 쪽)의 절반 크기와 같은 중첩이 사용된다. 예를 들어, (512 샘플 크기를 갖는) 서브프레임(612)의 중첩은 (256 샘플의 더 작은 크기를 갖는) 이전의 서브프레임(611)의 128 샘플이다. 다른 한편, 서브 프레임(615)(512 샘플 크기)은 이전의 서브프레임(614)(1024 샘플 크기)보다 작으므로, 이전의 서브프레임(614)의 256 샘플의 중첩을 사용한다. 대안적인 구현에서, 인코더는 이전 및 이후의 서브프레임과 상이한 중첩량(예를 들어, 더 작은 서브프레임 크기의 4분의 1 등등)으로 MLT를 사용할 수 있다. 이 예에서 알 수 있는 바와 같이, 3개의 서브프레임 크기의 선택은 더 큰 수의 이전, 현재 및 이후의 서브프레임 크기 조합이 생기게 함으로써, 서브프레임에 대한 겹침 변환 크기의 더 큰 순열을 생성한다.

[0089] 인코더에 의한 무손실 압축의 경우에, MLT는 바람직하게 가역 정수-정수 변환 구현을 갖는다. 다음 설명은 서브프레임 크기의 다양한 조합을 위해 이러한 가역 정수-정수 변환 구현이 얻어지는 방식에 관해 다룬다. 각 서브프레임 크기 조합에 대한 MLT는 서브프레임에 적용된 블록 변환(811-815)(도 8에 도시됨)과 함께, 서브프레임 들에 걸쳐 있는 중첩 윈도우 변환(711-714)(도 7에 도시됨)의 조합으로 분해함으로써 생성된다.

[0090] **IV. MLT(Modulated Lapped Transform)의 분해**

[0091] 다음 섹션은 MLT와 같은 특정 겹침 직교 변환을 이해하고 구현하기 위한 단순한 방법을 제공한다. MLT(MCLT의 실수부)는 오디오 코덱에서 일반적으로 사용되는 다음의 직교 변환(음의 부호는 임의로 붙인 것이지만, 이것이 (배울 외에는) 변환의 정의에 일치하기 때문에 추가됨)으로서 쓰여질 수 있다:

[0092] 
$$Y[k + Ml] = -\sqrt{\frac{2}{M}} \sum_{n=0}^{2M-1} x\left[n - \frac{M}{2} + Ml\right] \sin\left[\left(n + \frac{1}{2}\right) \frac{\pi}{2M}\right] \cos\left[\left(n + \frac{M+1}{2}\right) \left(k + \frac{1}{2}\right) \frac{\pi}{M}\right]$$

[0093] 상기 식에서, 입력 오디오 샘플은  $x\left[n - \frac{M}{2} + Ml\right]$  항에 의해 표시된다.  $\sin$  항은 윈도 함수인 반면에,  $\cos$  항은 변환의 기저 함수이다. 변수  $k$ 는 스펙트럼 지수이고,  $M$ 은 서브프레임 크기이며,  $l$ 은 서브프레임의 지수이다.

[0094]  $\cos(-\theta + \pi) = -\cos\theta$ 이기 때문에, 이것은 다음과 같이 된다:

$$\begin{aligned} & \cos\left[\left(M-1-n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ &= \cos\left[\left(-n-\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}+2M\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ &= \cos\left[\left(-n-\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}+(2k+1)\pi\right] \\ &= -\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \end{aligned}$$

[0095]

[0096] 이와 유사하게,  $\cos(-\theta+2\pi) = \cos\theta$ 이기 때문에, 우리는 또한 다음과 같이 된다는 것을 알고 있다:

$$\begin{aligned} & \cos\left[\left(\frac{3M}{2}-1-n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ &= \cos\left[\left(-n-\frac{3M}{2}-\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}+4M\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ &= \cos\left[\left(-n-\frac{3M}{2}-\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}+(4k+2)\pi\right] \\ &= \cos\left[\left(\frac{3M}{2}+n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \end{aligned}$$

[0097]

[0098] MLT 식에서 이러한 위의 두 관계를 적용하면, 우리는 다음과 같은 식을 얻는다:

$$\begin{aligned} Y[k+Ml] &= -\sqrt{\frac{2}{M}} \sum_{n=0}^{\frac{M}{2}-1} \left\{ \begin{aligned} & x[n+Ml] \sin\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] - \\ & x[-n-1+Ml] \sin\left[\left(-n+\frac{M-1}{2}\right)\frac{\pi}{2M}\right] \end{aligned} \right\} \cos\left[\left(n+M+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \\ & - \sqrt{\frac{2}{M}} \sum_{n=\frac{M}{2}}^{M-1} \left\{ \begin{aligned} & x[n+Ml] \sin\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] + \\ & x[2M-1-n+Ml] \sin\left[\left(-n+\frac{5M-1}{2}\right)\frac{\pi}{2M}\right] \end{aligned} \right\} \cos\left[\left(n+M+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \end{aligned}$$

[0099]

[0100]  $z[n]$  항이 다음과 같이 정의된다고 가정하자:

$$z[n+Ml] = x[n+Ml] \sin\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] - x[-n-1+Ml] \sin\left[\left(-n+\frac{M-1}{2}\right)\frac{\pi}{2M}\right],$$

for  $n=0, \dots, M/2-1$ , and

$$z[n+Ml] = x[n+Ml] \sin\left[\left(n+\frac{M+1}{2}\right)\frac{\pi}{2M}\right] + x[2M-1-n+Ml] \sin\left[\left(-n+\frac{5M-1}{2}\right)\frac{\pi}{2M}\right],$$

for  $n=M/2, \dots, M-1$ .

[0101]



[0102] 그러면, MLT 식을 다음과 같이 쓸 수 있는데:

$$\begin{aligned} Y[k + Ml] &= -\sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} z[n + Ml] \cos \left[ \left( n + M + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \\ &= -\sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} z[M - 1 - n + Ml] \cos \left[ \left( M - 1 - n + M + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right] \\ &= -\sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} z[M - 1 - n + Ml] \cos \left[ \left( -n - \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} + (2k + 1)\pi \right] \\ &= \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} z[M - 1 - n + Ml] \cos \left[ \left( n + \frac{1}{2} \right) \left( k + \frac{1}{2} \right) \frac{\pi}{M} \right], \end{aligned}$$

[0103]

[0104] 이것은 대칭이동된  $z[n+Ml]$ 의 IV형 DCT이다.

[0105] 이러한 다시 쓰여진 식이 보여주는 바와 같이, MLT는 두 개의 블록 기반의 변환으로 생각될 수 있다. 제1 블록 변환은  $x$ 에서  $z$ 까지의 중첩하는 1단 버터플라이고, 제2 블록 변환은 단순히 대칭이동된  $z$ 항의 블록 기반 DCT-IV이다.

[0106] MLT에 대한 2개의 인접한 블록들 사이의 중첩 영역은 블록  $l$ 의 제1 절반과 블록  $(l-1)$ 의 제2 절반으로 이루어진다.  $\sin(-\theta + \pi/2) = \cos \theta$ 이기 때문에, 우리는 이 중첩 영역에 대한  $z[n+Ml]$  항을 다음과 같이 쓸 수 있다:

$$\begin{aligned} z[n + Ml] &= x[n + Ml] \sin \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] - x[-n - 1 + Ml] \sin \left[ \left( -n + \frac{M-1}{2} \right) \frac{\pi}{2M} \right] \\ &= x[n + Ml] \sin \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] - x[-n - 1 + Ml] \cos \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] \end{aligned}$$

for  $n=0, \dots, M/2-1$ , and

$$z[n + Ml - M] = x[n + Ml - M] \sin \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] + x[2M - 1 - n + Ml - M] \sin \left[ \left( -n + \frac{5M-1}{2} \right) \frac{\pi}{2M} \right]$$

for  $n=M/2, \dots, M-1$ .

[0107]

[0108]  $n'=M-1-n$ 을 대입한 후에,  $\sin(\theta + \pi/2) = \cos \theta$ 이고  $\sin(-\theta + \pi) = \sin \theta$ 이기 때문에, 우리는 다음 식을 얻는다:

$$\begin{aligned} z[-1 - n + Ml] &= x[-n - 1 + Ml] \sin \left[ \left( -n + \frac{3M-1}{2} \right) \frac{\pi}{2M} \right] + x[n + Ml] \sin \left[ \left( n + \frac{3M-1}{2} \right) \frac{\pi}{2M} \right] \\ &= x[-n - 1 + Ml] \sin \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] + x[n + Ml] \cos \left[ \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M} \right] \end{aligned}$$

for  $n=0, \dots, M/2-1$ .

[0109]

[0110] 이것은 다음과 같은 단순 2x2 변환 블록과 동일한데:

$$\begin{bmatrix} z[n + Ml] \\ z[-1 - n + Ml] \end{bmatrix} = \begin{bmatrix} \sin \theta & -\cos \theta \\ \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} x[n + Ml] \\ x[-1 - n + Ml] \end{bmatrix}$$

[0111]

[0112] 여기에서,  $\theta = \left( n + \frac{M+1}{2} \right) \frac{\pi}{2M}$ ,  $n=0, \dots, M/2-1$ . 이다. 그 다음,  $n'=M/2-1-n$ 을 더 대입하면 다음 식이 제공되는데:

$$\begin{bmatrix} z[Ml + M/2 - 1 - n] \\ z[Ml - M/2 + n] \end{bmatrix} = \begin{bmatrix} \sin \theta & -\cos \theta \\ \cos \theta & \sin \theta \end{bmatrix} \begin{bmatrix} x[Ml + M/2 - 1 - n] \\ x[Ml - M/2 + n] \end{bmatrix}$$

[0113]

[0114] 여기서,  $n=0, \dots, M/2-1$  및  $\theta = \left(n + \frac{1}{2}\right) \frac{\pi}{2M}$  이고,  $\sin(-\theta + \pi/2) = \cos \theta$ 이고  $\cos(-\theta + \pi/2) = \sin \theta$ 라는 사실을 사용한다. 이 2x2 변환은 오디오 코덱의 중첩된 윈도 블록 변환(711-714)(도 7)으로서 사용될 수 있다. 변환은 각각의 중첩 영역을 1단 버터플라이로서 처리함으로써 가변 서브프레임 크기에 적용될 수 있다. 이 변환에서의 2x2 회전이 가역적으로 될 수 있기 때문에, 이 중첩 윈도우 변환은 용이하게 가역 변환으로 구현된다.

[0115] 서브프레임 크기가 (도 6에 도시된 예에서와 같이) 변할 때, MLT의 윈도 모양만이 변한다. 그러나, 윈도는 여전히 원래 윈도의 모든 대칭 속성에 따른다. 그러므로, 가변 크기 서브프레임에 적용할 때의 유일한 변경은 중첩하는 윈도 크기 M이다.

[0116] 더욱이, 이 방법은 도 6-8에 도시된 바와 같이, 임의의 직교 변환을 데이터 내의 제공된 블록에 단순히 적용한 다음에, 블록화 그리드를 이동시키고, 다른 직교 변환을 적용함으로써, 중첩된 직교 변환의 생성에 더욱 일반적으로 적용 가능하다.

[0117] **III. 일반 중첩 변환의 분해**

[0118] 특정 조건 하에서, 인접한 블록들 사이의 50% 이하 중첩을 갖는 임의의 변환은 2개의 블록 변환으로 분해될 수 있다. 정확한 필요 조건은 다음 유도에서 설명될 것이다. 50% 이하의 중첩을 갖는 블록에서 행해지는 겹침 변환을 고려해보자. 이러한 선형 변환은 행렬 내의 각 블록이  $N/2 \times N/2$  행렬인 행렬로서 쓰여질 수 있는데, N은 (가변 블록 크기가 허용되는 경우에) 변환 내의 최소 블록 크기이다. 예를 들어, 겹침 변환이 512, 1024 및 2048 크기 블록을 사용할 수 있게 허용된다고 가정하자. 그러면, 블록 크기가 512, 1024, 512 및 512인 경우에, 겹침 변환에 대한 변환 행렬은 다음과 같이 쓰여질 수 있다:

$$A = \begin{bmatrix} A_{00} & A_{01} & A_{02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{10} & A_{11} & A_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & 0 & 0 & 0 \\ 0 & A_{31} & A_{32} & A_{33} & A_{34} & A_{35} & A_{36} & 0 & 0 & 0 \\ 0 & A_{41} & A_{42} & A_{43} & A_{44} & A_{45} & A_{46} & 0 & 0 & 0 \\ 0 & A_{51} & A_{52} & A_{53} & A_{54} & A_{55} & A_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{65} & A_{66} & A_{67} & A_{68} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{75} & A_{76} & A_{77} & A_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{87} & A_{88} & A_{89} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{97} & A_{98} & A_{99} \end{bmatrix}$$

[0119]

[0120] 여기서, 각각의 부 행렬(sub-matrix)은 256x256 행렬이다.

[0121] 이 연산을 2단계로 나누는 것을 고려하면, 제1 단계는 중첩 연산자(C), 및 실제 블록 크기에서 블록 변환을 적용하는 다른 연산자(B)를 적용하는 것이다. 그러면, 우리는 이들 2개의 행렬을 다음과 같이 쓸 수 있다:

$$B = \begin{bmatrix} B_{00} & B_{01} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{10} & B_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{22} & B_{23} & B_{24} & B_{25} & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{32} & B_{33} & B_{34} & B_{35} & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{42} & B_{43} & B_{44} & B_{45} & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{52} & B_{53} & B_{54} & B_{55} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & B_{66} & B_{67} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & B_{76} & B_{77} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{88} & B_{89} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{98} & B_{99} \end{bmatrix}$$

[0122]

[0123] 및

$$C = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{11} & C_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{21} & C_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{55} & C_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{65} & C_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{77} & C_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{87} & C_{88} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}$$

[0124]

[0125] 목적은 A=BC가 되게 하는 B와 C를 찾는 것이다. A의 n번째 행이 B의 n번째 행에만 의존하기 때문에, 개별적으로 변환의 단일 블록을 고찰하는 것으로 충분하다. 예를 들어, 변환의 제3 블록은 다음과 같이 쓰여질 수 있다:

$$\begin{bmatrix} A_{65} & A_{66} & A_{67} & A_{68} \\ A_{75} & A_{76} & A_{77} & A_{78} \end{bmatrix} = \begin{bmatrix} B_{66} & B_{67} \\ B_{76} & B_{77} \end{bmatrix} \begin{bmatrix} C_{65} & C_{66} & 0 & 0 \\ 0 & 0 & C_{77} & C_{78} \end{bmatrix}$$

$$= \begin{bmatrix} B_{66}C_{65} & B_{66}C_{66} & B_{67}C_{77} & B_{67}C_{78} \\ B_{76}C_{65} & B_{76}C_{66} & B_{77}C_{77} & B_{77}C_{78} \end{bmatrix}$$

[0126]

[0127] 변환의 제3 블록의 제1 절반에 대한 해를 찾는 것은 다음 등식의 해를 찾는 것으로 단순화된다:

$$\begin{bmatrix} B_{66} \\ B_{76} \end{bmatrix} \begin{bmatrix} C_{65} & C_{66} \end{bmatrix} = \begin{bmatrix} A_{65} & A_{66} \\ A_{75} & A_{76} \end{bmatrix}$$

[0128]

[0129] 이것은 단지, 행렬이 다음과 같은 속성

$$A_{65} A_{75}^{-1} = A_{66} A_{76}^{-1},$$

[0130]

[0131] 을 갖는 경우에, 또는 제공된 j에 대한 모든 i에 대해  $A_{ij}=0$ 인 경우에 해를 구할 수 있다. 이것에 대한 다수의 해가 있다.  $B_{66}$ 을 어떤 임의의 전체 계수  $N/2 \times N/2$  행렬이라고 하자. 그러면, 우리는 다음을 얻는다:

$$C_{65} = B_{66}^{-1} A_{65}$$

$$C_{66} = B_{66}^{-1} A_{66}$$

$$B_{76} = A_{75} B_{66} A_{65}^{-1}$$

[0132]

[0133] 이 변환 블록 및 행렬 A 내의 그의 다른 변환 블록의 제2 절반은 이와 유사한 방식으로 풀이될 수 있다.

[0134] 이것은 특정 조건(즉, 속성  $A_{65} A_{75}^{-1} = A_{66} A_{76}^{-1}$ ) 하에서, 겹침 변환이 2개의 블록 변환의 곱으로 쓰여질 수 있는데, 그 하나는 중첩을 처리하고, 다른 하나는 단순 블록 변환이라는 것을 보여준다.

[0135] 따라서, 가변 크기 서브프레임을 갖는 오디오 신호의 서브프레임에 대한 겹침 변환의 가역적 정수-정수 구현은

크기 N의 서브프레임에 대해 직사각형 선형 변환 행렬 A 
$$= \begin{bmatrix} A_{i(j-1)} & A_{ij} & A_{i(j+1)} & A_{i(j+2)} \\ A_{(i+1)(j-1)} & A_{(i+1)j} & A_{(i+1)(j+1)} & A_{(i+1)(j+2)} \end{bmatrix}$$
 로서 겹침 변환의 표현을 형성함으로써 구해진다. 이 직사각형 선형 변환 행렬은  $N/2 \times N/2$  행렬 블록 원소로 구성되는데, i, j는 행렬 블록의 지수이다. 더욱이, 행렬은 주어진 n에 대한 모든 m에 대해

$A_{i(j-1)} A_{(i+1)(j-1)}^{-1} = A_{ij} A_{(i+1)j}^{-1}$  또는  $A_{mm}=0$ 이라는 속성을 갖는다. 직사각형 선형 변환 행렬 A는

$$\begin{bmatrix} C_{(i-1)(j-1)} & C_{(i-1)j} \\ C_{i(j-1)} & C_{ij} \end{bmatrix}$$
 형태의 중첩 윈도우 변환 행렬 C와 
$$\begin{bmatrix} B_{ij} & B_{i(j+1)} \\ B_{(i+1)j} & B_{(i+1)(j+1)} \end{bmatrix}$$
 형태의 블록 변환 행렬 B로 분해되는

데, 그 구성요소는 다음과 같은 관계식을 풀어서 얻어진다:

$$C_{i(j-1)} = B_{ij}^{-1} A_{i(j-1)}$$

$$C_{ij} = B_{ij}^{-1} A_{ij}$$

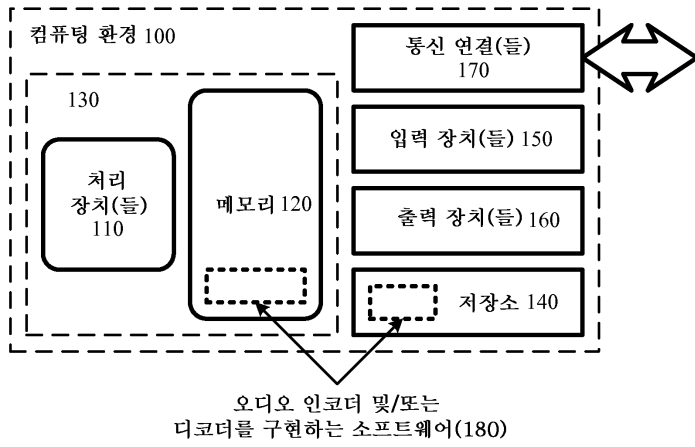
$$B_{(i+1)j} = A_{(i+1)(j-1)} B_{ij} A_{i(j-1)}^{-1}$$

[0136]

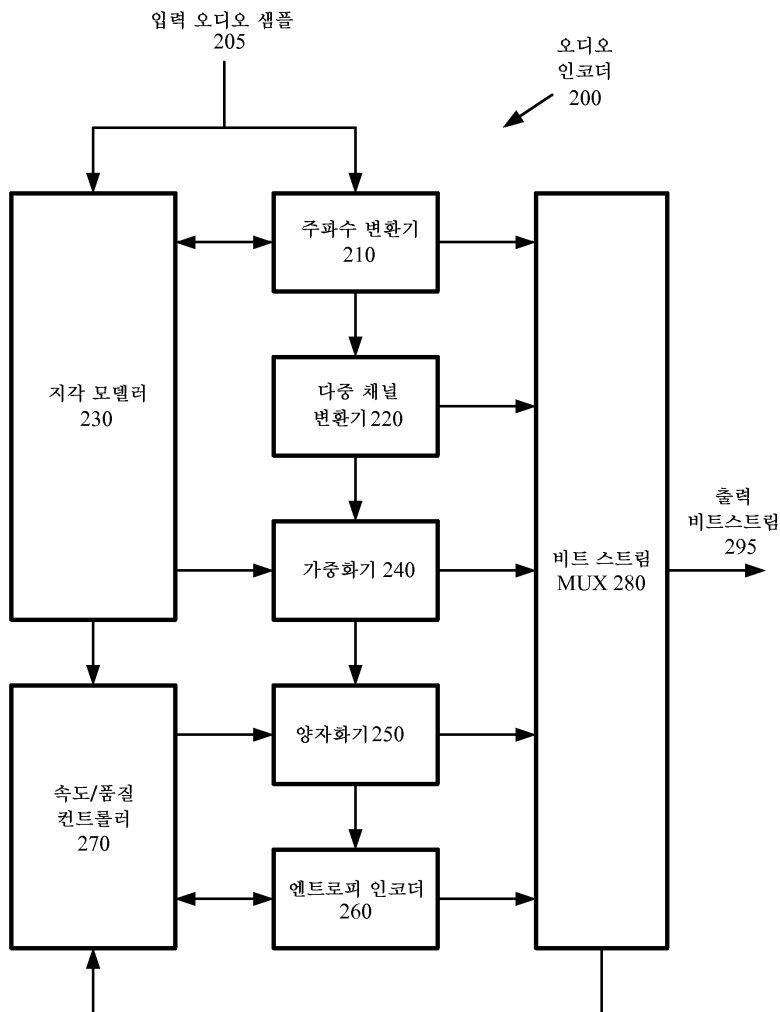
[0137] 본 발명의 원리가 적용될 수 있는 다수의 가능한 실시예를 고려하여, 우리는 다음 청구범위 및 그 등가물의 범위 및 정신에 속할 수 있는 모든 이러한 실시예를 우리의 발명으로서 청구한다.

도면

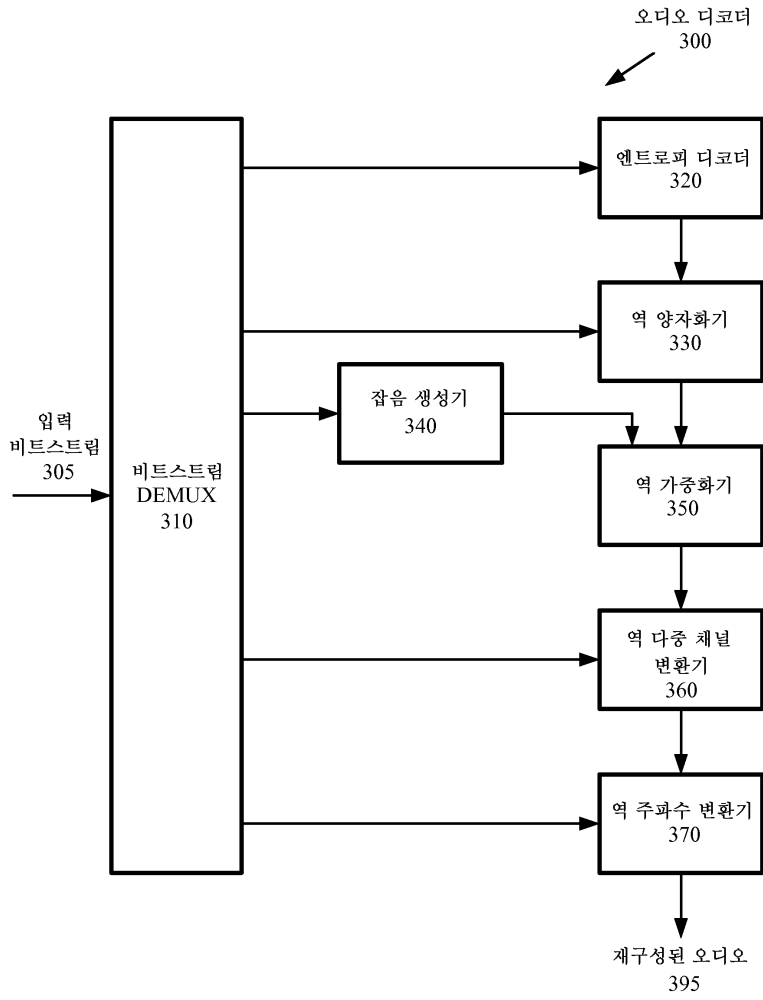
도면1



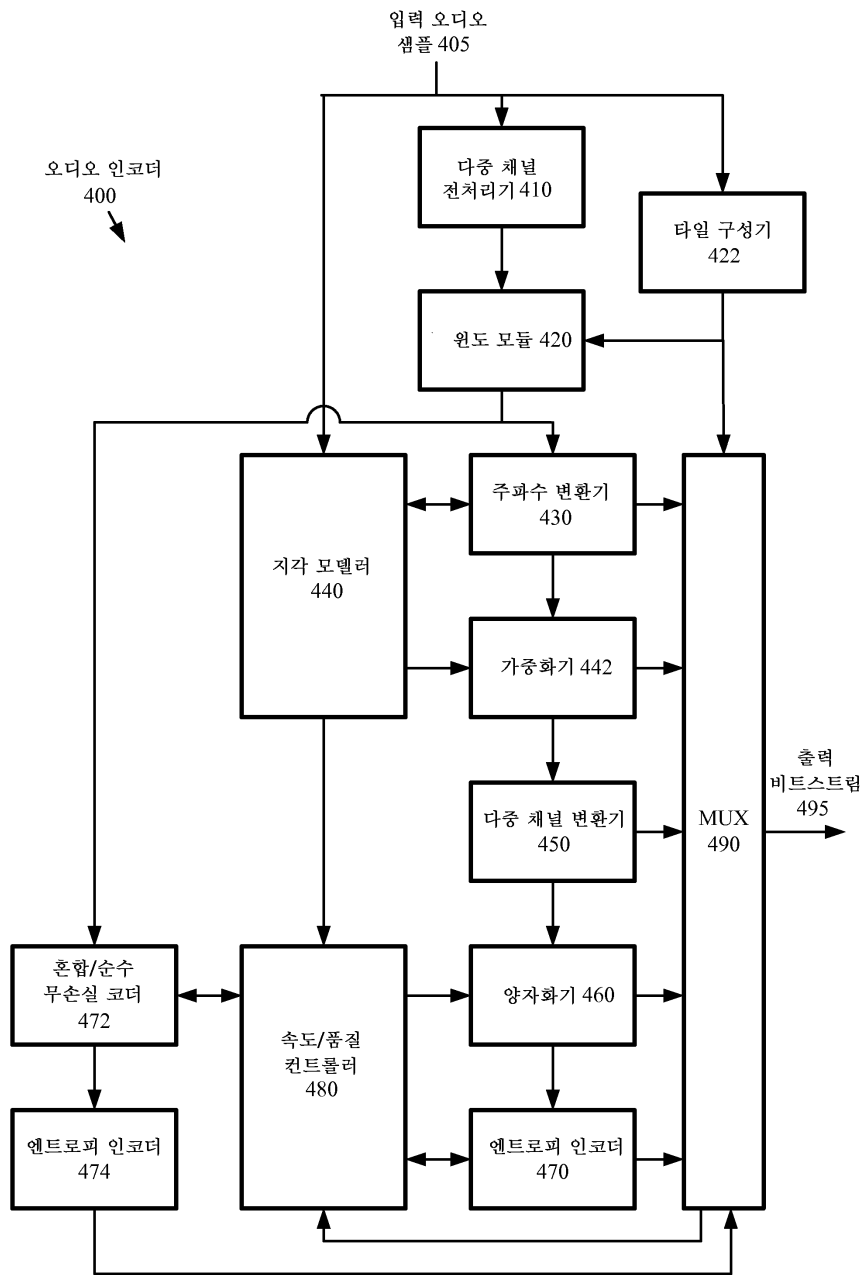
도면2



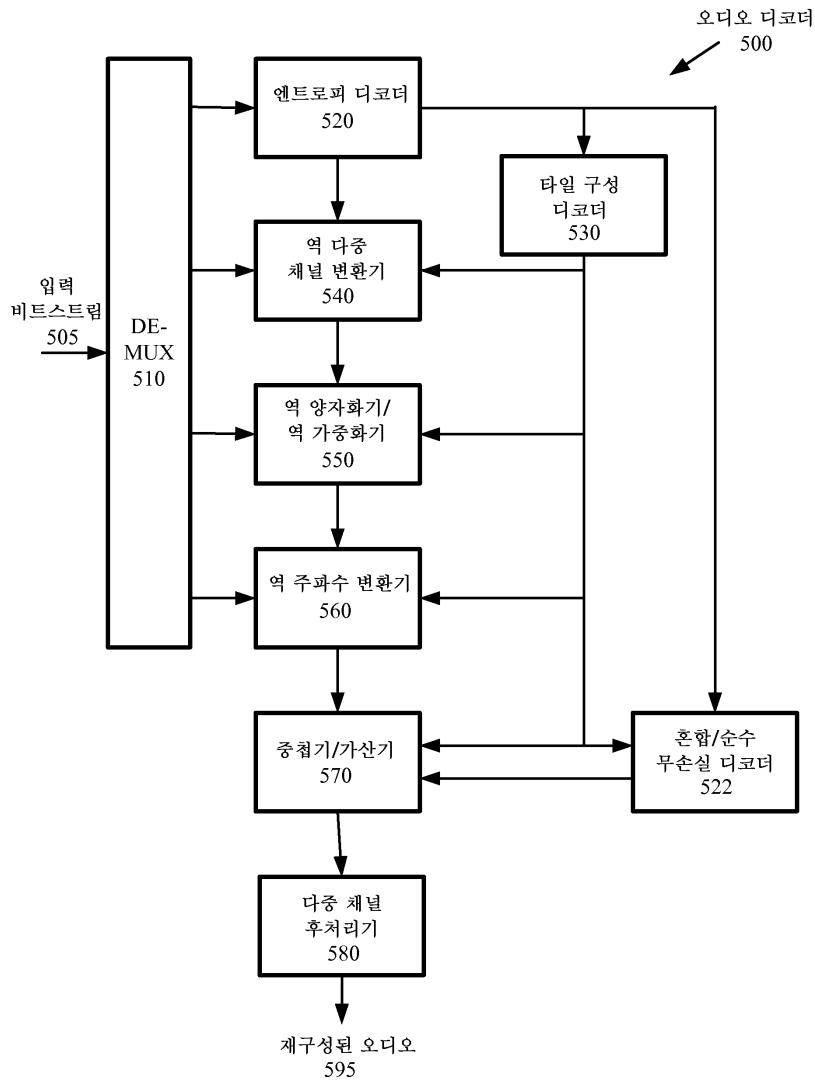
도면3



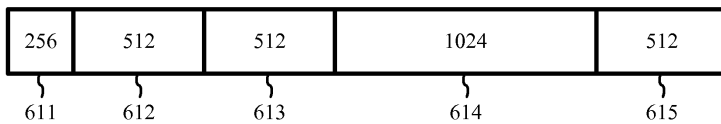
도면4



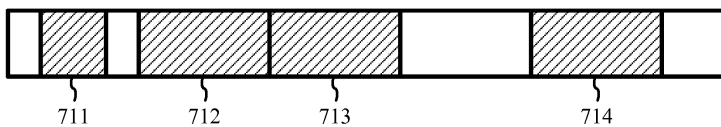
도면5



도면6



도면7





도면8

